INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING
Int. J. Adapt. Control Signal Process. 2014; **00**:1–14
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/acs

A unifying framework for the analysis of proportionate NLMS algorithms

B. Jelfs*and D. P. Mandic

Department of Electrical and Electronic Engineering, Imperial College London, UK

SUMMARY

Despite being a de facto standard in sparse adaptive filtering, the two most important members of the class of proportionate normalised least mean square (PNLMS) algorithms are introduced empirically. Our aim is to provide a unifying framework for the derivation of PNLMS algorithms and their variants with an adaptive step-size. These include algorithms with gradient adaptive learning rates and algorithms with adaptive regularisation parameters. Convergence analysis is provided for the proportionate least mean square (PLMS) algorithm in both the mean and mean square sense and bounds on its parameters are derived. An alternative, more insightful approach to the convergence analysis is also presented and is shown to provide an estimate of the optimal step-size of the PLMS. Incorporating the so obtained step-size into the PLMS gives the standard PNLMS together with a unified framework for introducing other adaptive learning rates. Simulations on benchmark sparse impulse responses support the approach. Copyright © 2014 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: Adaptive Filters; Convergence; Proportionate Normalized Least Mean Square; Sparse System Identification

1. INTRODUCTION

The least mean square (LMS) family of algorithms are a standard for the training of linear adaptive filters [1]. The aim of the LMS algorithm is to minimise the cost function

$$\mathcal{J}(k) = \frac{1}{2}e^2(k) \tag{1}$$

and is described by

$$\begin{aligned} e(k) &= d(k) - \mathbf{x}^T(k)\mathbf{w}(k), \\ \mathbf{w}(k+1) &= \mathbf{w}(k) + \mu e(k)\mathbf{x}(k), \end{aligned} \tag{2}$$

where $e(k)$ is the output error at time instant k, $d(k)$ the desired signal, and $\mathbf{x}(k) = [x(k), ..., x(k-N+1)]^T$ and $\mathbf{w}(k) = [w_1(k), ..., w_N(k)]^T$ are respectively the input signal and filter coefficient vector for a filter of length N. Critical to the performance of the algorithm is the step-size parameter μ which defines how fast the algorithm is converging towards the optimal solution. Where the optimal solution is not reached the step-size parameter also affects the misalignment between the algorithm solution and the optimal solution.

*Correspondence to: Beth Jelfs, Department of Electronic Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong. E-mail: beth.jelfs05@alumni.imperial.ac.uk

To facilitate operation in a nonstationary environment, that is, to allow the filter to adapt independently of the signal power in the tap input, the normalised LMS (NLMS) uses an adaptive step-size

$$\eta(k) = \frac{\mu}{\|\mathbf{x}(k)\|_2^2 + \varepsilon}, \tag{3}$$

where $\|\cdot\|_2$ is the Euclidean norm. For practical reasons, the regularisation parameter ε is included to prevent the weight update becoming unstable for input vectors comprising of near to zero elements. This would otherwise be interpreted as effectively a large learning rate $\eta(k)$ leading to divergence.

Due to the importance and wide range of practical applications of the LMS based algorithms, research into modifications of this class of algorithms has been a major topic in statistical and adaptive signal processing communities [1]. The selection of learning rate is critical to the performance of all LMS-type algorithms. Ideally, we desire an algorithm for which the speed of convergence is fast and the steady state error is small when operating in a stationary environment. In a nonstationary environment the algorithm is required to track the signal and as such the learning rate should change according to the dynamics of the input signal. A convenient way to improve the convergence of linear adaptive filters in nonstationary environments is to introduce a gradient adaptive step-size (GASS). Approaches with a "linear" gradient adaptive learning rate based on $\partial\mathcal{J}/\partial\mu$ include the algorithms by Benveniste *et al*. [2], Mathews and Xie [3], and Ang and Farhang [4]. Alternatively a "nonlinear" gradient adaptive step-size based on $\partial\mathcal{J}/\partial\varepsilon$, such as the Generalised Normalised Gradient Descent (GNGD) algorithm [5], can be employed.

1.1. Proportionate NLMS

Both the LMS and NLMS algorithms perform in a suboptimal manner in sparse environments [6,7] where the impulse response of an unknown system has a number of zero elements. As such, the development of adaptive filters designed specifically for such environments has become an increasingly large area of research, having particular application in echo and adaptive noise cancellation [8,9].

For operation in sparse environments, the proportionate NLMS (PNLMS) [7] develops on the NLMS algorithm to give an update which is proportional relative to the size of the filter coefficients. This is achieved by introducing a diagonal "tap selection matrix" $\mathbf{G}(k)$ within the coefficient update (2), giving the weight update [7]

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\frac{\mathbf{G}(k)e(k)\mathbf{x}(k)}{\|\mathbf{x}(k)\|_2^2}, \tag{4}$$

where

$$\mathbf{G}(k) = \mathrm{diag}[g_1(k), \ldots, g_N(k)]. \tag{5}$$

The diagonal elements of $\mathbf{G}(k)$ define the proportionate amounts that each coefficient is updated by and the elements $g_n(k)$ are given by

$$\begin{aligned}
\bar{\phi}(k) &= 1/N\sum_{n=1}^{N}\phi_n(k),\\
\phi_n(k) &= \max\left\{\rho\max\left[\delta, \|\mathbf{w}(k)\|_\infty\right], |w_n(k)|\right\},\\
g_n(k) &= \frac{\phi_n(k)}{\bar{\phi}(k)}, \qquad\qquad n = 1, \ldots, N
\end{aligned} \tag{6}$$

where the symbol $\|\cdot\|_\infty$ denotes the infinity norm.

A number of modifications of the PNLMS have been proposed [10, 11], however, the stability of the algorithm has received little attention. The existing approaches, such as Doroslovački and Deng's analysis [12], provide convergence proofs based on existing algorithms, such as steepest descent, and are not explicitly derived for PNLMS. Little attention is given to the fact that as the PNLMS is constrained to the stability limits of NLMS [7] and that it also inherits a problem frequently encountered with NLMS, namely that for an ill-conditioned tap input autocorrelation matrix or for processes exhibiting large dynamics, the filter becomes unstable [5].

The PNLMS algorithm in its original form has been introduced based on empirical evidence and subsequently most of its variants have also been designed in a similar manner. It would, however, be beneficial for both education and practical purposes, if the derivation and analysis of the class of PNLMS algorithms could be conducted based on a unified theoretical platform.

Our aim is therefore two-fold: firstly, following the results in [13], we provide a unified approach to the derivation of the class of PNLMS algorithms; secondly, expanding on these results we present adaptive step-size extensions of PNLMS algorithms based on $\frac{d\mathcal{J}}{d\mu} = 0$ [2–4] and also on the adaptation of the regularisation parameter ε [5]. Simulations on benchmark sparse systems support the analysis.

2. DERIVATION OF THE CLASS OF PNLMS ALGORITHMS

Originally Duttweiler [7] introduced the PNLMS algorithm (4) as a version of NLMS, however, to unify the analysis it is advantageous to start from the LMS. In the same vein as the Duttweiler result, we shall modify the LMS to suit sparse environments, thus providing a basis to derive the class of PNLMS algorithms in a generic way. The update of this "proportionate LMS" (PLMS) algorithm thus becomes

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\mathbf{G}(k)e(k)\mathbf{x}(k), \tag{7}$$

that is, the PLMS is equips the standard LMS with the "tap selective" term $\mathbf{G}(k)$. The use of tap selection also has a geometric justification, as for an N-tap LMS the weight update lives in $\mathbb{R}^N$ and the direction of the update is dominated by the largest element of the input vector $\mathbf{x}(k)$ [14]. We shall now provide a rigorous derivation of the standard PNLMS, based on the convergence analysis of PLMS in (7); this approach is supported by the original analysis performed by Duttweiler [7] which analyses a form of the PLMS algorithm with fixed gain distributors $\mathbf{G}$.

2.1. Convergence in the Mean

Assume without a loss in generality the desired response can be expressed as [15]

$$d(k) = \mathbf{x}^T(k)\mathbf{w}_o + q(k), \tag{8}$$

where $\mathbf{w}_o$ is the optimal weight vector and $q(k)$ is zero mean Gaussian noise with variance σ_q^2, uncorrelated with $\mathbf{x}(k)$. This allows us to describe the PLMS by

$$\begin{aligned} e(k) =& \mathbf{x}^T(k)\mathbf{w}_o + q(k) - \mathbf{x}^T(k)\mathbf{w}(k), \\ \mathbf{w}(k+1) =& \mathbf{w}(k) + \mu\mathbf{G}(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}_o - \mu\mathbf{G}(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k) + \mu q(k)\mathbf{G}(k)\mathbf{x}(k). \end{aligned} \tag{9}$$

Upon subtracting the optimal weight vector $\mathbf{w}_o$ from both sides, the weight error vector $\mathbf{v}(k) = \mathbf{w}(k) - \mathbf{w}_o$ can be expressed as

$$\mathbf{v}(k+1) = \mathbf{v}(k) - \mu\mathbf{G}(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{v}(k) + \mu q(k)\mathbf{G}(k)\mathbf{x}(k). \tag{10}$$

We can now apply the statistical expectation operator and employ the independence assumptions[†] to yield

$$E[\mathbf{v}(k+1)] = \left(\mathbf{I} - \mu\mathbf{G}(k)E[\mathbf{x}(k)\mathbf{x}^T(k)]\right) E[\mathbf{v}(k)] + \mu\mathbf{G}(k)E[q(k)\mathbf{x}(k)],$$

[†] Namely that the input signal and filter coefficient vectors are zero mean, stationary, jointly normal and with finite moments; the successive increments of tap weights are independent of one another and the error and input vector sequences are statistically independent of one another [1].

DOI: 10.1002/acs

$$=\big(\mathbf{I}-\mu\mathbf{G}(k)\mathbf{R}\big)E[\mathbf{v}(k)], \tag{11}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{R}=E[\mathbf{x}(k)\mathbf{x}(k)^T]$ is the autocorrelation matrix of the input vector. Since the correlation matrix $\mathbf{R}$ is symmetric and positive semidefinite, it has the following decomposition

$$\mathbf{R}=\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T, \tag{12}$$

where $\mathbf{\Lambda}=diag(\lambda_1,\lambda_2,\ldots,\lambda_N)$ are the eigenvalues and $\mathbf{Q}$ is the orthogonal matrix of the corresponding eigenvectors. A rotation of the weight error vector $\mathbf{v}(k)$ by the eigenmatrix $\mathbf{Q}$, that is, $\mathbf{v}'(k)=\mathbf{Q}^T\mathbf{v}(k)$, gives

$$\mathbf{v}'(k+1)=\big(\mathbf{I}-\mu\mathbf{G}(k)\mathbf{\Lambda}\big)\mathbf{v}'(k). \tag{13}$$

As $(\mathbf{I}-\mu\mathbf{G}(k)\mathbf{\Lambda})$ is diagonal, every element of $\mathbf{v}'(k)$ evolves independently and converges to zero if and only if, for all the eigenvalues $|1-\mu g_n\lambda_n|<1$. Therefore, the PLMS converges only if it converges for the maximum mode of convergence, that is, for λ_{max}. Since $\lambda_{max}\leq\sum$(diagonal elements of $\mathbf{R}$) = tr[$\mathbf{R}$], the condition for the convergence in the mean of the PLMS becomes

$$0<\mu<\frac{2}{\mathrm{tr}[\mathbf{G}(k)\mathbf{R}]}. \tag{14}$$

Note that the term $\mathrm{tr}[\mathbf{G}(k)\mathbf{R}]$ is equivalent to $\mathbf{x}^T(k)\mathbf{G}(k)\mathbf{x}(k)$, however, for simplicity, for a white i.i.d. input for which $\mathbf{R}=\sigma_x^2\mathbf{I}$, and using the identity $\mathrm{tr}[AB]\leq\mathrm{tr}[A]\mathrm{tr}[B]$ and $\mathrm{tr}[\mathbf{G}(k)]=N$, we have

$$0<\mu<\frac{2}{N\sigma_x^2}. \tag{15}$$

2.2. *Convergence in the Mean Square*

To converge in the mean square the algorithm must firstly converge in the mean, that is, (14) must be satisfied. Based on the output error (9), we can arrive at the mean square error by squaring and taking the expectation of both sides, to give

$$E[e^2(k)]=\sigma_q^2+E\left[(\mathbf{x}^T(k)\mathbf{v}(k))^2\right]-2E\left[q(k)\mathbf{x}^T(k)\mathbf{v}(k)\right]. \tag{16}$$

Using the independence assumptions and $\mathbf{x}^T(k)\mathbf{v}(k)=\mathbf{v}^T(k)\mathbf{x}(k)$ and defining the weight error vector correlation matrix as $\mathbf{K}(k)=E[\mathbf{v}(k)\mathbf{v}^T(k)]$, results in

$$\begin{aligned}E\left[\left(\mathbf{x}^T(k)\mathbf{v}(k)\right)^2\right]&=\mathrm{tr}\left[E\left[\mathbf{v}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{v}(k)\right]\right]\\&=\mathrm{tr}\left[E\left[\mathbf{v}^T(k)E\left[\mathbf{x}(k)\mathbf{x}^T(k)\right]\mathbf{v}(k)\right]\right]\\&=E\left[\mathrm{tr}\left[\mathbf{v}^T(k)\mathbf{R}\mathbf{v}(k)\right]\right]\\&=E\left[\mathrm{tr}\left[\mathbf{v}(k)\mathbf{v}^T(k)\mathbf{R}\right]\right]\\&=\mathrm{tr}\left[E\left[\mathbf{v}(k)\mathbf{v}^T(k)\right]\mathbf{R}\right]\\&=\mathrm{tr}\left[\mathbf{R}\mathbf{K}(k)\right].\end{aligned} \tag{17}$$

Thus, the expected value of the squared error becomes

$$\begin{aligned}\xi(k)=E[e^2(k)]&=\xi_{min}+\xi_{EMSE}(k)\\&=\sigma_q^2+\mathrm{tr}[\mathbf{R}\mathbf{K}(k)]\end{aligned}$$

where the final term of (16) disappears due to the independence assumptions [1], ξ_{min} is the minimum mean square error defined by the power of the noise, σ_q^2, and $\xi_{EMSE}(k)$ is the excess mean square error. The excess mean square error is a result of the filter coefficients fluctuating around their optimum values as they begin to converge. Using again the rotation (12) and $\mathbf{K}'(k) = \mathbf{Q}^T\mathbf{K}(k)\mathbf{Q}$ gives

$$\xi(k) = \sigma_q^2 + \mathrm{tr}[\mathbf{K}'(k)\mathbf{\Lambda}]. \tag{18}$$

Since matrix $\mathbf{\Lambda}$ is diagonal, we have

$$\xi(k) = \sigma_q^2 + \sum_{i=1}^{N} \lambda_i \kappa'_{ii}(k), \tag{19}$$

where κ'_{ii} are the diagonal elements of $\mathbf{K}'$.

It is convenient to assess the mean square performance of an algorithm in terms of the misadjustment

$$M = \frac{\xi_{EMSE}}{\xi_{min}} = \frac{\xi_{EMSE}}{\sigma_q^2}, \tag{20}$$

which following the approach from [16, 17] can be shown to be

$$M = \frac{\sum_{i=1}^{N} \frac{\mu g_i \lambda_i}{2(1-\mu g_i(k)\lambda_i)}}{1 - \sum_{i=1}^{N} \frac{\mu g_i(k)\lambda_i}{2(1-\mu g_i(k)\lambda_i)}} \tag{21}$$

$$= \mu \frac{\frac{1}{2}\mathrm{tr}[\mathbf{G}(k)\mathbf{R}]}{1 - \frac{1}{2}\mu\mathrm{tr}[\mathbf{G}(k)\mathbf{R}]}. \tag{22}$$

Thus, for a white i.i.d. input, the mean square error $\xi(k)$ converges asymptotically to $\xi(\infty) = \mathcal{J}_{min} = \sigma_q^2$ for

$$0 < \mu < \frac{2}{\mathrm{tr}[\mathbf{G}(k)\mathbf{R}]} = \frac{2}{\sigma_x^2 N}. \tag{23}$$

which gives the bound on the step-size of PLMS.

2.3. *Introducing PNLMS via Normalisation of PLMS*

We have shown that for convergence of PLMS in both the mean and the mean square, the bound on the step-size is given by

$$0 < \mu < \frac{2}{\mathbf{x}^T(k)\mathbf{G}(k)\mathbf{x}^{(}k)}. \tag{24}$$

Incorporating this step-size into the PLMS (7) gives the normalised PLMS (PNLMS)

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\mu}{\mathbf{x}^T(k)\mathbf{G}(k)\mathbf{x}(k)}\mathbf{G}(k)e(k)\mathbf{x}(k), \tag{25}$$

where $0 < \mu < 2$.

As the expansion of $\mathbf{K}'(k)$ to obtain (21) is not straightforward, an alternative method is to recast the PNLMS into the optimisation task performed by NLMS type algorithms, that is, to minimise the a posteriori error $\tilde{e}(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k+1)$, as opposed to LMS which minimises the a priori error $e(k)$. We aim to arrive at the PNLMS by estimating the range of values of μ for which $|\tilde{e}(k)| < |e(k)|$. This is achieved (following the approach from [18]) by performing a first order Taylor series expansion (TSE) of $|\tilde{e}(k)|^2$ around $|e(k)|^2$, that is

$$|\tilde{e}(k)|^2 = |e(k)|^2 + \sum_{i=1}^{N} \frac{\partial |e(k)|^2}{\partial w_i(k)} \Delta w_i(k) \tag{26}$$

The partial derivative in (26) can be obtained from (2) as

$$\frac{\partial |e(k)|^2}{\partial w_i(k)} = -2e(k)x(k-i+1) = -2e(k)x_i(k), \qquad i = 1, 2, \ldots, N \tag{27}$$

while the update $\Delta w_i(k)$ in (7), is given by

$$\Delta w_i(k) = \mu e(k) g_i(k) x_i(k). \qquad i = 1, 2, \ldots, N \tag{28}$$

A substitution of (27)–(28) into (26) yields

$$\begin{aligned} |\tilde{e}(k)|^2 =& |e(k)|^2 - 2\mu \left[e(k) \sum_{i=1}^{N} x_i(k) \right] \left[e(k) \sum_{i=1}^{N} g_i(k) x_i(k) \right] \\ =& |e(k)|^2 - 2\mu |e(k)|^2 \mathbf{x}^T(k)\mathbf{G}(k)\mathbf{x}(k) \\ =& |e(k)|^2 \left[1 - 2\mu \mathbf{x}^T(k)\mathbf{G}(k)\mathbf{x}(k) \right]. \end{aligned} \tag{29}$$

For the output error to vanish towards zero as $k \to \infty$ we require

$$|\tilde{e}(k)|^2 \leq |e(k)|^2 \left[1 - 2\mu \mathbf{x}^T(k)\mathbf{G}(k)\mathbf{x}(k) \right]. \tag{30}$$

As the squared error terms are non-negative this will occur if and only if

$$\left| 1 - 2\mu \mathbf{x}^T(k)\mathbf{G}(k)\mathbf{x}(k) \right| < 1, \tag{31}$$

resulting in the following bounds on the range of the step-size

$$0 < \mu \leq \frac{1}{\mathbf{x}^T(k)\mathbf{G}(k)\mathbf{x}(k))}. \tag{32}$$

These bounds are optimal, providing the first order TSE gives a good approximation of the a posteriori error $\tilde{e}(k)$. From (32), to minimise the a posteriori error $\tilde{e}(k)$ and equip the proportionate LMS with an optimal learning rate we have

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \frac{\mathbf{G}(k)e(k)\mathbf{x}(k)}{\mathbf{x}^T(k)\mathbf{G}(k)\mathbf{x}(k)}, \tag{33}$$

The original PNLMS was introduced empirically and uses the standard NLMS update to obtain its optimal step-size. Notice that the update presented here has precisely the form of what has become the standard version of PNLMS [11].

2.3.1. Comparison of Different Formulations of PNLMS The above formulation of the PNLMS, whilst having an update of $\mathbf{w}(k)$ in line with that of the standard version of the PNLMS, does differ in one point. In the original version of the PNLMS the update of the proportionate term $\mathbf{G}(k)$ is given by (6) and has the form

$$\bar{\phi}(k) = 1/N \sum_{n=1}^{N} \phi_n(k), \tag{34}$$

resulting in $\mathrm{tr}[\mathbf{G}(k)] = N$. However, in the more commonly used version of the PNLMS the term $\bar{\phi}$ assumes the form

$$\bar{\phi}(k) = \sum_{n=1}^{N} \phi_n(k), \tag{35}$$

which results in $\mathrm{tr}[\mathbf{G}(k)] = 1$. In this case the bound on the step-size (23) becomes

$$0 < \mu < \frac{2}{\mathrm{tr}[\mathbf{R}]} = \frac{2}{\mathbf{x}^T(k)\mathbf{x}(k)}, \tag{36}$$

and results in an update of $\mathbf{w}(k)$ in the form of the original PNLMS (4). As such, the version of the update of $\mathbf{w}(k)$ depends on the update of $\mathbf{G}(k)$ used; the version of the PNLMS presented here is therefore an optimal version and not a perfect replica of either the original Duttweiler version or the more standard version.

Regarding the discrepancies between the PNLMS formulations, it has been shown in [19] that when including a regularisation parameter in the standard version of the PNLMS, as with the NLMS, the PNLMS version is a scaled version of the NLMS regularisation by a factor of N. One advantage of using this formulation of the PNLMS is that it allows for the regularisation irrespective of the proportionality component of the algorithm. In this way, the regularisation of both the NLMS and PNLMS are functions of the filter length, N, the signal variance, σ_x^2, and the signal to noise ratio (SNR), making the choice of regularisation critical in low SNR conditions. This form of regularisation has been shown to provide stability in stationary environments for which adaptive versions which also allow operation in nonstationary environments will be presented in Section 3.2.

3. UNIFYING APPROACH TO GASS ALGORITHMS FOR PNLMS

Now that we have obtined the PNLMS as a result of an optimisation procedure, it is natural to equip this approach with fast and robust learning, by considering adaptive learning rates. The optimal learning rate obtained from the TSE of the a posteriori error $\tilde{e}(k)$ in (29) uses only the first term of the expansion, that is, the partial derivatives with respect to $w_i, i = 1, \dots, N$, and thus the TSE (26) can be expanded to include the higher order terms ($h.o.t.$) as

$$|\tilde{e}(k)|^2 = |e(k)|^2 - 2\mu|e(k)|^2\mathbf{x}^T(k)\mathbf{G}(k)\mathbf{x}(k) + h.o.t.(k). \tag{37}$$

Algorithms which take into account higher order statistics can provide better modelling of the signal distribution for non-Gaussian inputs [20]. To account for the exclusion of the neglected higher order terms of TSE (26), we next provide an insight into the extent to which the higher order terms influence the algorithm.

The class of gradient adaptive step-size (GASS) algorithms include the algorithm proposed in [3] which is based upon a gradient adaptation of the learning rate of the LMS from (2), or in the form $\frac{\partial \mathcal{J}(k)}{\partial \mu}$ [3], where $\mathcal{J}(k)$ is the cost function (1). This algorithm is essentially a simplified version of the Benveniste *et al*. algorithm [2], where time varying filtered versions of the instantaneous gradients of the cost function with respect to the learning rate are replaced by their instantaneous values. In the algorithm of Ang and Farhang [4], this time variant filtering of the instantaneous gradients is replaced by a low pass filter with a fixed coefficient. To arrive at this class of algorithms for PNLMS, we shall express the higher order terms from (37) as

$$h.o.t.(k) = \theta(k)e(k). \tag{38}$$

On the other hand, the GNGD class of algorithms [5], make the parameter ε in the step-size of the NLMS in (3) gradient adaptive based on $\frac{\partial \mathcal{J}(k)}{\partial \varepsilon(k-1)}$, which for the PNLMS results in the higher order terms in the form

$$h.o.t.(k) = -\mu e(k)\varepsilon(k). \tag{39}$$

By setting (37) to zero and solving for μ, we can now express the optimal step-sizes for the Mathews and Xie type GASS algorithm [3] for PNLMS as

$$\mu_{opt} = \frac{\eta(k)}{\mathbf{x}^T(k)\mathbf{G}(k)\mathbf{x}(k)} \tag{40}$$

and the GNGD type step-size update for the PNLMS as

$$\mu_{opt} = \frac{1}{\mathbf{x}^T(k)\mathbf{G}(k)\mathbf{x}(k) + \varepsilon(k)}. \tag{41}$$

Notice from (40) and (41) that there is a fundamental difference between the variable step-size algorithms with a “linear” multiplicative adaptive factor (GASS framework) and GNGD, which employs a “nonlinear” update of the adaptive learning rate. The exact derivation of the update in either case depends crucially on the weight sensitivities with respect to the adaptive term within the step-size.

3.1. Linear Update of the Adaptive Step-Size

In this case, the term $h.o.t.(k)$ is written as $\theta(k)e(k)$, so that a gradient adaptive step-size algorithm similar to that of the Mathews and Xie algorithm for the LMS becomes

$$\begin{aligned}
\mathbf{w}(k+1) &= \mathbf{w}(k) + \frac{\eta(k)}{\mathbf{x}^T(k)\mathbf{G}(k)\mathbf{x}(k)}\mathbf{G}(k)e(k)\mathbf{x}(k), \\
\eta(k+1) &= \eta(k) - \beta\frac{\partial\mathcal{J}(k)}{\partial\eta(k-1)}, \\
\frac{\partial\mathcal{J}(k)}{\partial\eta(k-1)} &= \frac{\partial\mathcal{J}(k)}{\partial e(k)}\frac{\partial e(k)}{\partial\mathbf{w}(k)}\frac{\partial\mathbf{w}(k)}{\partial\eta(k-1)}, \\
\eta(k+1) &= \eta(k)+\beta\frac{e(k)e(k-1)\mathbf{x}^T(k)\mathbf{G}(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)},
\end{aligned} \tag{42}$$

where β is a small positive constant. In practice, learning rates need to be smaller than the optimal ones[‡] and the term $\eta(k)$ from above is often replaced by a small constant μ which multiplies an adaptive parameter, for instance $\eta(k) = \mu\vartheta(k)$.

The result in (42) is an approximation of the rigorous analysis provided by Benveniste *et al*. [2] where,

$$\eta(k+1) = \eta(k) + \beta e(k)\mathbf{x}^T(k)\gamma(k) \tag{43}$$

and

$$\gamma(k) = [1 - \eta(k-1)]\,\gamma(k-1) + \frac{\mathbf{G}(k-1)e(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)}, \tag{44}$$

or that of Ang and Farhang [4]

$$\gamma(k) = \alpha\gamma(k-1) + \frac{\mathbf{G}(k-1)e(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)}, \tag{45}$$

where α is a constant such that $\alpha < 1$. For further detail see Appendix A.

3.2. Nonlinear Update of the Adaptive Step-Size

In the second case, the term $h.o.t.$ from (37) is written as $h.o.t. = -\mu e(k)\varepsilon(k)$, which allows us to arrive at the GNGD type PNLMS algorithm, where the variation of the time varying μ_{opt} is governed in a nonlinear manner. The weight update of this algorithm is derived starting from

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\mu}{\mathbf{x}(k)^T\mathbf{G}(k)\mathbf{x}(k) + \varepsilon(k)}\mathbf{G}(k)e(k)\mathbf{x}(k). \tag{46}$$

[‡] The optimal learning rates are usually derived for white inputs and based on the independence assumptions.

The parameter ε in (46) is then made gradient adaptive as

$$\begin{aligned}
\varepsilon(k+1) =&\varepsilon(k) - \beta \nabla_{\varepsilon(k-1)} \mathcal{J}(k), \\
\frac{\partial \mathcal{J}(k)}{\partial \varepsilon(k-1)} =& \frac{\partial \mathcal{J}(k)}{\partial e(k)} \frac{\partial e(k)}{\partial \mathbf{w}(k)} \frac{\partial \mathbf{w}(k)}{\partial \varepsilon(k-1)}, \\
=&\mu \frac{e(k)e(k-1)\mathbf{x}^T\mathbf{G}(k)\mathbf{x}(k-1)}{\left[\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)+\varepsilon(k-1)\right]^2}, \\
\varepsilon(k+1) =&\varepsilon(k) - \beta\mu \frac{e(k)e(k-1)\mathbf{x}^T(k)\mathbf{G}(k-1)\mathbf{x}(k-1)}{\left[\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)+\varepsilon(k-1)\right]^2},
\end{aligned} \tag{47}$$

where β is a small positive constant. The adaptation of the time–varying term in the denominator of the above gradient adaptive learning rate compensates for the deficiencies in the derivation of the step-size of the NLMS [18] and hence the PNLMS. The complete derivation can be found in Appendix B.

4. SIMULATIONS

To illustrate the performance of the algorithms derived within the proposed GASS framework, we ran simulations in a system identification setting and for a range of the algorithm parameters. Learning curves were produced using the normalised misalignment in dB, given by $10\log_{10}\|\mathbf{w}_{\text{opt}} - \mathbf{w}\|_2^2/\|\mathbf{w}_{\text{opt}}\|_2^2$, averaged over 100 independent trials, where $\mathbf{w}_{\text{opt}} = [w_{1\,\text{opt}}, ..., w_{N\,\text{opt}}]^T$ is the optimal filter coefficient vector. For all the algorithms, the parameters ρ and δ were set to the recommended values of $\rho = 5/N$ and $\delta = 0.01$ [7]. The regularisation parameter ε was set to $\varepsilon = 0.01$ while for the adaptive-ε algorithm $\varepsilon(0) = 0.01$, whereas for the Ang & Farhang type algorithm the learning rate $\alpha = 0.9$. The sparse system under consideration was the benchmark system analysed in [6], a 100-tap channel with four nonzero taps located in positions $[1, 30, 35, 85]$. For each trial, the input given to the channel was white Gaussian and the noise signal was an independent white Gaussian noise with the SNR set to 20dB; the initial estimates of the filter coefficients $\mathbf{w}(0)$ were selected randomly in order to provide a range of different learning curves.

The performance of the standard PNLMS compared with that of the proposed algorithms for a step-size of μ and $\eta(0) = 0.1$ is shown in Fig. 1. For all the linear learning rate updates the step-size was $\beta = 0.01$. In this case, the Benveniste and Farhang PNLMS type algorithms offered faster rates of convergence, but the PNLMS and adaptive ε algorithms had a smaller steady state error, with the Mathews algorithm in between them. Figure 2 shows convergence curves for the same parameter settings, except that for the linear learning rates the parameter β was set to $\beta = 0.001$. A comparison of the performances of algorithms with "linear" learning rates in Fig. 2, with those in Fig. 1 highlights the sensitivity issues associated with the linear learning rates and also the selection of their parameters.

To illustrate the behaviour of the algorithms in typical critical operating conditions (simulating the effect of close to zero inputs[§]), the value of μ and $\eta(0)$ was increased to 1.95, at which point PNLMS is on the limit of stability. As shown in Fig. 3, with the value of $\beta = 0.01$ the Benveniste algorithm did not converge, and the Farhang algorithm offered a smaller steady state error than the Mathews algorithm. Observe that when the algorithms were operating in critical conditions, the performance of the algorithm with an adaptive regularisation factor was still much improved compared to the PNLMS. Figure 4 illustrates that by reducing the value of β to $\beta = 0.001$ causes the Benveniste algorithm to converge, however, the performance was in line with that of the adaptive ε algorithm.

[§] Notice from (40)–(41) that as $\mathbf{x}(k)\mathbf{G}(k)\mathbf{x}(k) \to 0$ then effectively $\mu(k) \to \infty$. By making μ and $\eta(0)$ large we achieve the same effect, allowing a convenient comparison of the fixed, linear, and variable-ε methods.

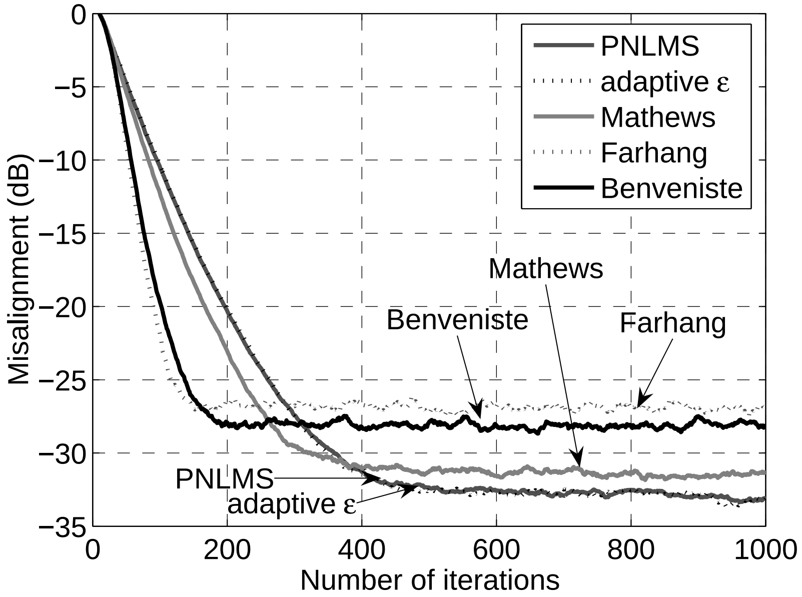

Figure 1. Performance comparison of the GASS PNLMS algorithms with the standard PNLMS for $\mu = 0.1$ and $\beta = 0.01$

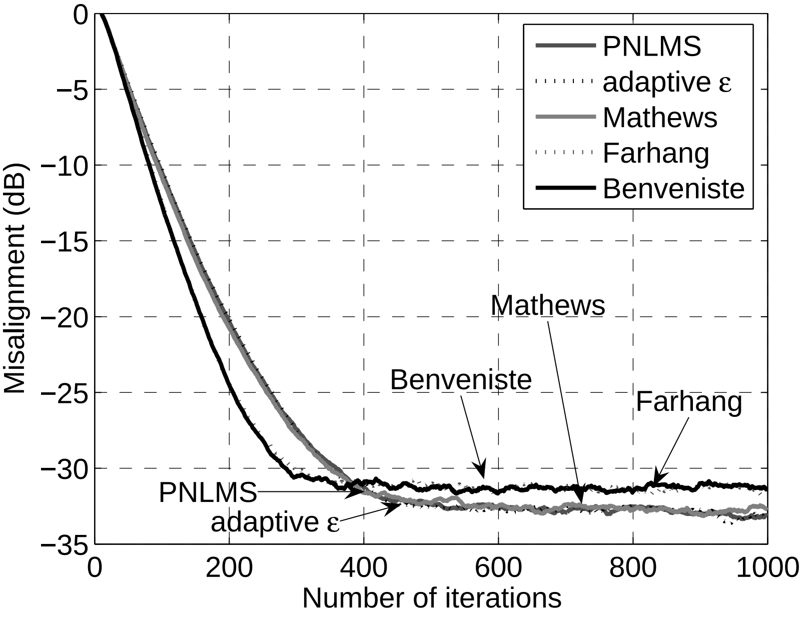

Figure 2. Performance comparison of the GASS algorithms with the PNLMS for $\mu = 0.1$ and $\beta = 0.001$

5. CONCLUSIONS

We have provided a unified approach to the derivation of the class of PNLMS algorithms, starting from a standard LMS through to the PNLMS and its adaptive step-size variants. This has been achieved in a generic way, and has allowed us to introduce two classes of adaptive step-size algorithms within the same framework. The algorithm by Benveniste *et al.*, although the most

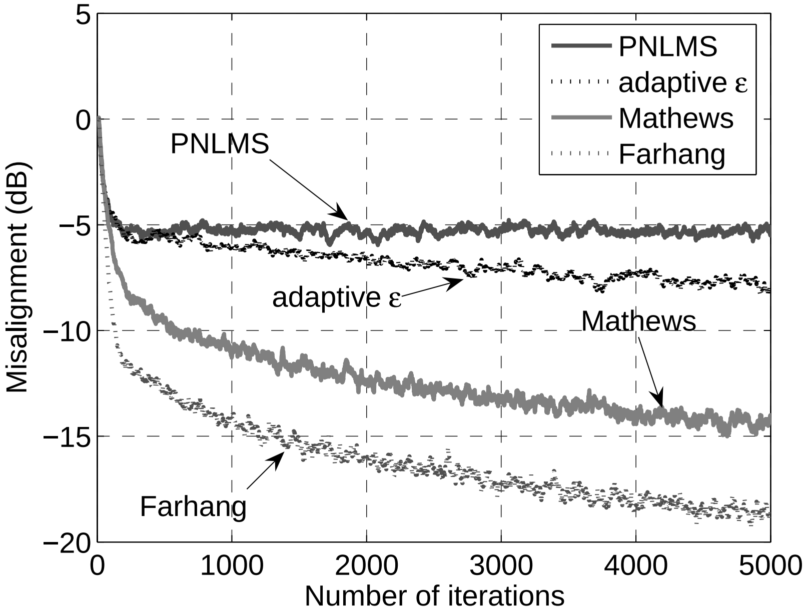

Figure 3. Performance comparison of the GASS algorithms with the PNLMS for $\mu = 1.95$ and $\beta = 0.01$

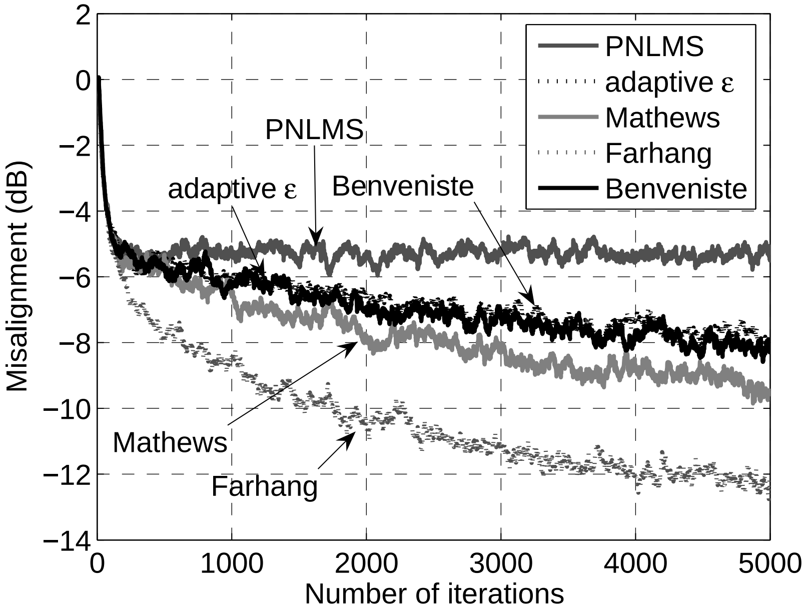

Figure 4. Performance comparison of the GASS algorithms with the PNLMS for $\mu = 1.95$ and $\beta = 0.001$

mathematically rigorous, when combined with the empirically derived PNLMS, has been shown to lack robustness. In comparison, the other "linear" adaptive learning rate PNLMS algorithms, Ang and Farhang's and Mathews and Xie's have exhibited enhanced robustness. The "nonlinear" GNGD type adaptive-ε algorithm has proved to be significantly easier to tune and less sensitive to initial conditions and when used in an ill-conditioned environment it always remains stable. This

DOI: 10.1002/acs

framework can be used as a unifying framework for the derivation and analysis of any of the class of PNLMS algorithms.

A. DERIVATION OF ADAPTIVE STEP-SIZE ALGORITHM

Following the approach from [2], we briefly sketch a rigorous derivation of adaptive step-size PNLMS algorithms. In this approach, the term $\frac{\partial \mathbf{w}(k)}{\partial \eta(k-1)} = \gamma(k)$ is derived based on the PNLMS update (33). The step-size update in gradient adaptive step-size algorithms now becomes

$$\eta(k+1) = \eta(k) + \beta e(k)\mathbf{x}(k)\gamma(k) \tag{48}$$

and $\gamma(k)$ is obtained from a matrix equation [2, 4]

$$\begin{aligned}\gamma(k) =& \frac{\partial \mathbf{w}(k-1)}{\partial \eta(k-1)} + \frac{\partial \eta(k-1)}{\partial \eta(k-1)} \cdot \frac{\mathbf{G}(k-1)e(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)} \\ &+ \frac{\partial \mathbf{G}(k-1)}{\partial \eta(k-1)} \cdot \frac{\eta(k-1)e(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)} + \frac{\partial e(k-1)}{\partial \eta(k-1)} \cdot \frac{\eta(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)} \\ &+ \frac{\partial[\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)]^{-1}}{\partial \eta(k-1)} \cdot \eta(k-1)\mathbf{G}(k-1)e(k-1)\mathbf{x}(k-1).\end{aligned} \tag{49}$$

Assuming that for small η, $\eta(k-1) \approx \eta(k) \therefore \frac{\partial \mathbf{w}(k)}{\partial \eta(k-1)} = \frac{\partial \mathbf{w}(k)}{\partial \eta(k)} = \gamma(k)$ and $\gamma(k-1) = \frac{\partial \mathbf{w}(k-1)}{\partial \eta(k-1)}$ and substituting $\lambda(k)$ with $\frac{\partial \mathbf{G}(k)}{\partial \eta(k)}$, we have

$$\begin{aligned}\gamma(k) =& \gamma(k-1) + \frac{\mathbf{G}(k-1)e(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)} + \lambda(k-1)\frac{\eta(k-1)e(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)} \\ &- \mathbf{x}^T(k-1)\gamma(k-1)\frac{\eta(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)} \\ &- \frac{\mathbf{x}^T(k-1)\lambda(k-1)\mathbf{x}(k-1)}{[\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)]^2} \cdot \eta(k-1)\mathbf{G}(k-1)e(k-1)\mathbf{x}(k-1), \\ \gamma(k) =& \left[1 - \eta(k-1)\frac{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)}\right] \cdot \gamma(k-1) + \frac{\mathbf{G}(k-1)e(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)} \\ &+ \left[1 - \frac{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)}\right] \cdot \frac{\eta(k-1)e(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)}\lambda(k-1), \\ \gamma(k) =& \left[1 - \eta(k-1)\right]\gamma(k-1) + \frac{\mathbf{G}(k-1)e(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)}.\end{aligned} \tag{50}$$

In the algorithm [4], the time varying term in the square brackets from the above is replaced by a constant $\alpha < 1$, thus, performing fixed parameter low pass filtering. The algorithm from [3], which we have addressed above, is therefore a special case of [2] and [4] when the parameter α is set to $\alpha = 0$.

B. DERIVATION OF ADAPTIVE REGULARISATION ALGORITHM

In this case, $\frac{\partial \mathbf{w}(k)}{\partial \varepsilon(k-1)} = \gamma(k)$ and the update of the regularisation factor now becomes

$$\varepsilon(k+1) = \varepsilon(k) + \beta e(k)\mathbf{x}(k)\gamma(k) \tag{51}$$

whereas $\gamma(k)$ is obtained from

$$
\begin{aligned}
\gamma_i(k) =& \frac{\partial \mathbf{w}(k-1)}{\partial \varepsilon(k-1)} + \frac{\partial \mathbf{G}(k-1)}{\partial \varepsilon(k-1)} \cdot \mu \frac{e(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1) + \varepsilon(k-1)} \\
&+ \frac{\partial e(k-1)}{\partial \varepsilon(k-1)} \cdot \mu \frac{\mathbf{G}(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1) + \varepsilon(k-1)} \\
&+ \frac{\partial [\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1) + \varepsilon(k-1)]^{-1}}{\partial \varepsilon(k-1)} \cdot \mu \mathbf{G}(k-1)e(k-1)\mathbf{x}(k-1), \\
\gamma(k) =& \gamma(k-1) + \lambda(k-1) \cdot \mu \frac{e(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1) + \varepsilon(k-1)} \\
&- \mathbf{x}^T(k-1)\gamma(k-1) \cdot \mu \frac{\mathbf{G}(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1) + \varepsilon(k-1)} \\
&- \frac{x^T(k-1)\lambda(k-1)\mathbf{x}(k-1) + 1}{[\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1) + \varepsilon(k-1)]^2} \cdot \mu \mathbf{G}(k-1)e(k-1)\mathbf{x}(k-1), \\
\gamma(k) =& \left[1 - \mu \frac{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1) + \varepsilon(k-1)}\right] \cdot \gamma(k-1) \\
&- \mu \frac{\mathbf{G}(k-1)e(k-1)\mathbf{x}(k-1)}{[\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1) + \varepsilon(k-1)]^2} \\
&+ \left[1 - \frac{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1) + \varepsilon(k-1)}\right] \cdot \\
&\cdot \mu \frac{e(k-1)\mathbf{x}(k-1)}{\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1) + \varepsilon(k-1)} \lambda(k-1).
\end{aligned}
\tag{52}
$$

Allowing $\gamma(k-1) = \frac{\partial \mathbf{w}(k-1)}{\partial \varepsilon(k-1)}$ and $\lambda(k) = \frac{\partial \mathbf{G}(k)}{\partial \varepsilon(k)}$, for small ε gives

$$
\gamma(k) = [1-\mu]\,\gamma(k-1) - \mu \frac{\mathbf{G}(k-1)e(k-1)\mathbf{x}(k-1)}{[\mathbf{x}^T(k-1)\mathbf{G}(k-1)\mathbf{x}(k-1) + \varepsilon(k-1)]^2}. \tag{53}
$$

The equation derived in (47) is a simplified version of this strict update.

REFERENCES

1. Sayed A. *Fundamentals of Adaptive Filtering*. Wiley, 2003.
2. Benveniste A, Metivier M, Priouret P. *Adaptive Algorithms and Stochastic Approximations*. Springer–Verlag, 1990.
3. Mathews V, Xie Z. A stochastic gradient adaptive filter with gradient adaptive step size. *IEEE Transactions on Signal Processing* 1993; **41**(6):2075–2087.
4. Ang WP, Farhang-Boroujeny B. A new class of gradient adaptive step–size LMS algorithms. *IEEE Transactions on Signal Processing* 2001; **49**(4):805–810.
5. Mandic D. A generalized normalized gradient descent algorithm. *IEEE Signal Processing Letters* 2004; **11**(2):115–118.
6. Martin R, Sethares W, Williamson R, Johnson C Jr. Exploiting sparsity in adaptive filters. *IEEE Transactions on Signal Processing* 2002; **50**(8):1883–1894.
7. Duttweiler D. Proportionate normalized least–mean–squares adaptation in echo cancelers. *IEEE Transactions on Speech and Audio Processing* 2000; **8**(5):508–518.
8. Boukis C, Constantinides A. Hierarchical filters in a collaborative filtering framework for system identification and knowledge retrieval. *Signal Processing Techniques for Knowledge Extraction and Information*, Mandic D, Golz M, Kuh A, Obradovic D, Tanaka T (eds.). Springer, 2008.
9. Zhang Y, Chambers J, Kendrick P, Cox T, Li F. A combined blind source separation and adaptive noise cancellation scheme with potential application in blind acoustic parameter extraction. *Neurocomputing* 2008; **71**(10–12):2127–2139.
10. Deng H, Doroslovacki M. Proportionate adaptive algorithms for network echo cancellation. *IEEE Transactions on Signal Processing* 2006; **54**(5):1794–1803.
11. Benesty J, Gay S. An improved PNLMS algorithm. *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, 2002; 1881–1884.
12. Doroslovački M, Deng H. On convergence of proportionate-type NLMS adaptive algorithms. *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 3, 2006; 105–108.

DOI: 10.1002/acs

13. Jelfs B, Mandic D, Cichocki A. A unifying approach to the derivation of the class of PNLMS algorithms. *15th International Conference on Digital Signal Processing (DSP)*, 2007; 35–38.
14. Soria E, Calpe J, Chambers J, Martinez M, Camps G, Guerrero J. A novel approach to introducing adaptive filters based on the LMS algorithms and its variants. *IEEE Transactions on Eduction* 2004; **47**(1):127–133.
15. Douglas S, Pan W. Exact expectation analysis of the LMS adaptive filter. *IEEE Transactions on Signal Processing* 1995; **43**(12):2863–2871.
16. Feuer A, Weinstein E. Convergence analysis of LMS filters with uncorrelated Gaussian data. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 1985; **ASSP–33**(1):222–230.
17. Farhang-Boroujeny B. *Adaptive Filters: Theory and Applications*. Wiley, 1998.
18. Soria-Olivas E, Calpe-Maravilla J, Guerrero-Martinez J, Martinez-Sober M, Espi-Lopez J. An easy demonstration of the optimum value of the adaptation constant in the LMS algorithm. *IEEE Transactions on Education* 1998; **41**(1):81.
19. Benesty J, Paleologu C, Ciochină S. On regularization in adaptive filtering. *IEEE Transactions on Audio, Speech, and Language Processing* 2011; **19**(6):1734–1742.
20. Barrros A, Principe J, Takeuchi Y, Ohnishi N. Using non-linear even functions for error minimization in adaptive filters. *Neurocomputing* 2006; **70**(1–3):9–13.

DOI: 10.1002/acs