A Roundoff Error Analysis of the Normalized LMS Algorithm

Pi Sheng Chang and Alan N. Willson, Jr.

Electrical Engineering Department
University of California, Los Angeles
Los Angeles, CA 90095
e-mail: pschang@ee.ucla.edu, willson@ee.ucla.edu

Abstract

This paper describes an analysis of the Normalized LMS algorithm under finite word-length effects. It is shown that, using the Averaging Principle, it is possible to derive a good approximation for the total MSE in steady-state. Implementation issues are considered. It is shown that, in general, the quantization of the division operation will not cause the performance of the algorithm to deteriorate. An example using a look-up table for the division operation is analyzed and simulations are shown to support the analysis.

1 Introduction

Normalized LMS (NLMS) has been known to perform better than LMS in several applications. Recent work has shown that its analysis is not straightforward and its performance can be quite different from LMS [1, 2, 6], but almost no study has been done of its performance when fixed point arithmetic is used. Here we complement the analysis made in [3] for LMS, by applying the same approach to the NLMS. Some approximations are introduced in order to find a closed-form expression for the total MSE in steady-state. Furthermore, simulations were done to show quantitatively the amount of discrepancy when such approximations are used.

1.1 The Normalized LMS Algorithm

The approach used here is similar to the one presented in [3]. The same notation and assumptions are used. By assumptions we refer to the usual *Fundamental Assumptions* and the *Principle of Orthogonality* [4]. The basic NLMS algorithm is given by the following equation:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \frac{\mu e_n \mathbf{x}_n}{||\mathbf{x}_n||^2} \tag{1}$$

where e_n is the estimation error, $\mathbf{w}_n$ is the vector of tap weights at instant n, $\mathbf{x}_n$ is the input vector of dimension N, and $||\mathbf{x}_n||^2 = \mathbf{x}_n^T\mathbf{x}_n$ is the square of the Euclidian vector norm. Furthermore, we consider the division operation as a scalar-valued function of $\mathbf{x}_n$ so that it can be better handled later, then (1) can be rewritten as

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu e_n \mathbf{x}_n f(\mathbf{x}_n). \tag{2}$$

2 Fixed-Point Arithmetic

It is assumed that rounding is used as quantization in the fixed-point arithmetic. The quantizer is modeled as an additive error, and it's notation is given by $Q[a] = a + \epsilon$, where ϵ is the quantization error. The quantization error associated with a B-plus-sign-bit representation is assumed to have uniform distribution with variance $\sigma^2 = 2^{-2B}/12$. Data samples are represented by B_d-plus-sign-bit numbers with variance σ_d^2 and weight coefficients are represented by B_c-plus-sign-bit numbers with variance σ_c^2. Unprimed symbols represent infinite precision quantities while primed symbols represent finite precision quantities. No overflow is assumed, thus additions don't introduce error, only multiplications. The new input sequence and weight vector in finite precision arithmetic becomes

$$\mathbf{x}'_n = \mathbf{x}_n + \boldsymbol{\alpha}_n \tag{3}$$

$$\mathbf{w}'_n = \mathbf{w}_n + \boldsymbol{\rho}_n. \tag{4}$$

2.1 The Model

Fig. 1 shows the block diagram of an adaptive filter. The power-of-two scaling factor a is used in order to avoid overflow in the weight vector $\mathbf{w}_n$ while tracking the input signal, as described in [3]. The scaled desired sequence y'_n and the computed output of the filter $\hat{y}'_n$ are given by

$$\begin{aligned} y'_n &= y_n + \beta_n & (5)\\ \hat{y}'_n &= Q[\mathbf{w'}_n^T \mathbf{x}'_n] \\ &= \mathbf{w}_n^T\mathbf{x}_n + \boldsymbol{\rho}_n^T\mathbf{x}_n + \mathbf{w}_n^T\boldsymbol{\alpha}_n + \eta_n & (6)\end{aligned}$$

The variance of η_n, where $\sigma_\eta^2 = c\sigma_d^2$, depends on how the inner product is implemented [3]. If only the result

*This work was supported in part by CNPq, Brazil and in part by the Office of Naval Research under Grant N00014-95-1-0231.

1058-6393/96 $5.00 © 1996 IEEE
Proceedings of ASILOMAR-29

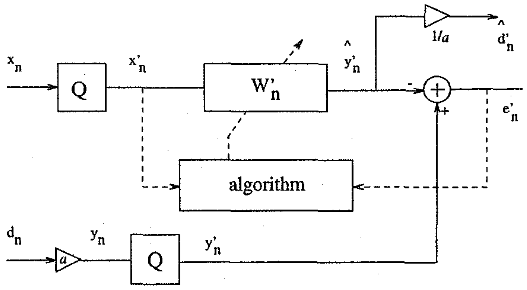

Fig. 1. Model for the adaptive filter as shown in [3].

is quantized, $c = 1$, otherwise, if every internal multiplication is quantized, then $c = N$. The estimation error e'_n is [3]

$$\begin{aligned} e'_n &= y'_n - \hat{y}'_n \\ &= y_n - \mathbf{w}_n^T \mathbf{x}_n - \boldsymbol{\rho}_n^T \mathbf{x}_n - \mathbf{w}_n^T \boldsymbol{\alpha}_n + \beta_n - \eta_n \quad (7) \end{aligned}$$

and the total output error is

$$d_n - \hat{d}'_n = \frac{1}{a} e_n - \frac{1}{a} (\boldsymbol{\rho}_n^T \mathbf{x}_n + \mathbf{w}_n^T \boldsymbol{\alpha}_n + \eta_n) \qquad (8)$$

where

$$e_n = y_n - \mathbf{w}_n^T \mathbf{x}_n. \qquad (9)$$

2.2 Total Output Mean Square Error

The term e_n/a is the estimation error of the infinite precision algorithm and the mean squared value associated with this term is given by (see Appendix A)

$$\xi_{inf.\ prec.} = \frac{\xi_{min}}{1 - \mu \sum_{i=1}^{N} \frac{\lambda_i}{2tr\mathbf{R} - \mu \lambda_i}} \qquad (10)$$

where $\xi_{min} = E[(d_n - \mathbf{w}^{*T} \mathbf{x}_n)^2]$ and $\lambda_i, i = 1, .., N$, are the eigenvalues of $\mathbf{R}$. The mean squared value of the arithmetic error is equal to [3]

$$\xi_{ar} = \frac{1}{a^2} \{ E[(\boldsymbol{\rho}_n^T \mathbf{x}_n)^2] + E[(\mathbf{w}_n^T \boldsymbol{\alpha}_n)^2] + E[\eta_n^2] \}. \qquad (11)$$

In steady-state, the second term of expression (11) is equal to (see Appendix B)

$$E[(\mathbf{w}_n^T \boldsymbol{\alpha}_n)^2] = E[\mathbf{w}_n^T \mathbf{w}_n] \sigma_d^2 =$$

$$\left(||\mathbf{w}^*||^2 + \frac{\mu \xi_{min}}{1 - \mu \sum_{i=1}^{N} \frac{\lambda_i}{2tr\mathbf{R} - \mu \lambda_i}} \sum_{i=1}^{N} \frac{1}{2tr\mathbf{R} - \mu \lambda_i} \right) \sigma_d^2 \qquad (12)$$

The expression for $E[(\boldsymbol{\rho}_n^T \mathbf{x}_n)^2]$ in steady-state is derived in a similar way to that in [3], where we have

$$E[(\boldsymbol{\rho}_n^T \mathbf{x}_n)^2] = tr(E[\boldsymbol{\rho}_n \boldsymbol{\rho}_n^T] \mathbf{R}). \qquad (13)$$

Rewriting (7), using $\zeta_n = \beta_n - \eta_n$ and (9) we have

$$e'_n = e_n - \boldsymbol{\rho}_n^T \mathbf{x}_n - \mathbf{w}_n^T \boldsymbol{\alpha}_n + \zeta_n \qquad (14)$$

where ζ_n has variance $(1 + c)\sigma_d^2$. The weight update according to the NLMS algorithm is

$$\begin{aligned} \mathbf{w'}_{n+1} &= \mathbf{w'}_n + Q[\mu \mathbf{x'}_n e'_n f(\mathbf{x}_n)] \\ &= \mathbf{w'}_n + \mu \mathbf{x'}_n e'_n f(\mathbf{x}_n) + \boldsymbol{\sigma}_n \\ &= \mathbf{w'}_n + \mu \mathbf{x}_n e_n f(\mathbf{x}_n) - \mu \mathbf{x}_n \mathbf{x}_n^T \boldsymbol{\rho}_n f(\mathbf{x}_n) - \end{aligned}$$

$$\mu \mathbf{x}_n \mathbf{w}_n^T \boldsymbol{\alpha}_n f(\mathbf{x}_n) + \mu \mathbf{x}_n \zeta_n f(\mathbf{x}_n) + \mu \boldsymbol{\alpha}_n e_n f(\mathbf{x}_n) + \boldsymbol{\sigma}_n \qquad (15)$$

Substituting (4) into the previous expression, we have

$$\boldsymbol{\rho}_{n+1} = \mathbf{F}_n \boldsymbol{\rho}_n + \mathbf{b}_n \qquad (16)$$

where

$$\mathbf{F}_n = I - \mu \mathbf{x}_n \mathbf{x}_n^T f(\mathbf{x}_n) \qquad (17)$$

and

$$\begin{aligned} \mathbf{b}_n = &-\mu \mathbf{x}_n \mathbf{w}_n^T \boldsymbol{\alpha}_n f(\mathbf{x}_n) + \mu \mathbf{x}_n \zeta_n f(\mathbf{x}_n) \\ &+ \mu \boldsymbol{\alpha}_n e_n f(\mathbf{x}_n) + \sigma_n. \qquad (18) \end{aligned}$$

Using $\mathbf{P}_n$ as $E[\rho_n \rho_n^T]$ and $\mathbf{Q}_n$ as $E[\mathbf{b}_n \mathbf{b}_n^T]$ we get

$$\mathbf{P}_{n+1} = E[\mathbf{F}_n \mathbf{P}_n \mathbf{F}_n] + \mathbf{Q}_n \qquad (19)$$

and

$$\mathbf{Q}_n = \mu^2 E[\mathbf{w}_n^T \mathbf{w}_n] \sigma_d^2 E[f(\mathbf{x}_n)^2 \mathbf{x}_n \mathbf{x}_n^T] + \mu^2 (1 + c) \cdot$$

$$\sigma_d^2 E[f(\mathbf{x}_n)^2 \mathbf{x}_n \mathbf{x}_n^T] + \mu^2 E[e_n^2] \sigma_d^2 E[f(\mathbf{x}_n)^2] \mathbf{I} + \sigma_c^2 \mathbf{I}. \quad (20)$$

In the steady-state, as $n \to \infty$,

$$\mathbf{Q}_\infty = \mu^2 ||\mathbf{w}^*||^2 \sigma_d^2 E[f(\mathbf{x}_n)^2 \mathbf{x}_n \mathbf{x}_n^T] + \mu^2 (1 + c) \cdot$$

$$\sigma_d^2 E[f(\mathbf{x}_n)^2 \mathbf{x}_n \mathbf{x}_n^T] + \mu^2 \xi_{min} \sigma_d^2 E[f(\mathbf{x}_n)^2] \mathbf{I} + \sigma_c^2 \mathbf{I}. \quad (21)$$

It would be more correct to use the exact value for $E[\mathbf{w}_n^T \mathbf{w}_n]$ as in (12), but here we have used the same approximation made in [3], so that both algorthms can better be compared. Still from [3], we can get the recursion

$$\begin{aligned} \mathbf{P}_{n+1} = \mathbf{P}_n - \mu (E[f(\mathbf{x}_n) \mathbf{x}_n \mathbf{x}_n^T] \mathbf{P}_n + \mathbf{P}_n E[f(\mathbf{x}_n) \mathbf{x}_n \mathbf{x}_n^T]) \\ + \mu^2 E[f(\mathbf{x}_n)^2 \mathbf{x}_n \mathbf{x}_n^T] tr[\mathbf{R} \mathbf{P}_n] + \mathbf{Q}_n. \qquad (22) \end{aligned}$$

Using the *Averaging Principle* (see discussion in Appendix A), it is possible to simplify the expression above by separating the $f(\mathbf{x}_n)$ terms. Thus, we have

$$\begin{aligned} \mathbf{P}_{n+1} = \mathbf{P}_n - \mu (E[f(\mathbf{x}_n)] \mathbf{R} \mathbf{P}_n + \mathbf{P}_n E[f(\mathbf{x}_n)] \mathbf{R}) \\ + \mu^2 E[f(\mathbf{x}_n)^2] \mathbf{R} tr[\mathbf{R} \mathbf{P}_n] + \mathbf{Q}_n. \qquad (23) \end{aligned}$$

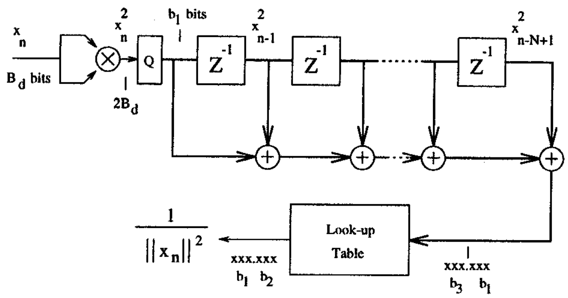

Fig. 2. Example of implementation of $f(\mathbf{x}_n)$.

In the steady-state, $\mathbf{P}_{n+1}$ becomes equal to $\mathbf{P}_n$, in the limit, so we can write

$$tr(\mathbf{R}\mathbf{P}_n) = tr(E[\rho_n \rho_n^T]\mathbf{R})$$
$$= \frac{tr\mathbf{Q}_n}{2\mu E[f(\mathbf{x}_n)] - \mu^2 E[f(\mathbf{x}_n)^2] tr\mathbf{R}}. \tag{24}$$

Putting everything together, we have

$$\xi = \frac{\xi_{min}}{1 - \mu \sum_{i=1}^{N} \frac{\lambda_i}{2tr\mathbf{R} - \mu\lambda_i}} + \frac{1}{a^2}\cdot$$
$$\left[\frac{N\sigma_c^2 + \mu^2[(1 + c + \|\mathbf{w}^*\|^2) tr\mathbf{R} + \xi_{min} N] E[f(\mathbf{x}_n)^2]\sigma_d^2}{2\mu E[f(\mathbf{x}_n)] - \mu^2 E[f(\mathbf{x}_n)^2] tr\mathbf{R}}\right.$$
$$+ (\|\mathbf{w}^*\|^2 + \frac{\mu \xi_{min}}{1 - \mu \sum_{i=1}^{N} \frac{\lambda_i}{2tr\mathbf{R} - \mu\lambda_i}}\cdot$$
$$\left.\sum_{i=1}^{N} \frac{1}{2tr\mathbf{R} - \mu\lambda_i})\sigma_d^2 + c\sigma_d^2\right]. \tag{25}$$

3 Example of Implementation

Since we have to define a specific implementation in order to find $E[f(\mathbf{x}_n)]$ and $E[f(\mathbf{x}_n)^2]$, consider, as an example, the implementation of $f(\mathbf{x}_n)$ as shown in Fig. 2. For this particular example we have

$$f(\mathbf{x}_n) = \frac{1}{\|\mathbf{x}_n\|^2 + q1_n} + q2_n \tag{26}$$

where the quantization noise $q1_n$ is due to the quantization before entering the look-up table and $q2_n$ is due to the quantization of the inverse function stored in the look-up table. Assume that the value after the multiplier is rounded to b_1 bits, and that the look-up table was built using rounding as quantization, then the following approximation is valid

$$E[f(\mathbf{x}_n)] \approx \frac{1}{tr\mathbf{R}} \tag{27}$$
$$E[f(\mathbf{x}_n)^2] \approx \frac{1}{(tr\mathbf{R})^2 + \sigma_{q1}^2} + \sigma_{q2}^2 \tag{28}$$

TABLE I. FIR filter coefficients used for the system to be identified.

i	w_i^*	i	w_i^*
0	3.422070×10^{-3}	10	9.878996×10^{-2}
1	8.640357×10^{-3}	11	9.353092×10^{-2}
2	1.671130×10^{-2}	12	8.371592×10^{-2}
3	2.767372×10^{-2}	13	7.061812×10^{-2}
4	4.105275×10^{-2}	14	5.584495×10^{-2}
5	5.584494×10^{-2}	15	4.105275×10^{-2}
6	7.061811×10^{-2}	16	2.767372×10^{-2}
7	8.371592×10^{-2}	17	1.671130×10^{-2}
8	9.353091×10^{-2}	18	8.640357×10^{-3}
9	9.878996×10^{-2}	19	3.422070×10^{-3}

TABLE II. Values of ξ ($\times 10^{-6}$), theoretical and simulated, for NLMS algorithm.

μ	B_d	B_c	b_1	b_2	Simul.	Theor.
1	inf. precision				5.98	6.02
	12	12	8	8	6.33	6.48
	12	12	4	4	6.37	6.49
	12	16	8	8	6.00	6.04
0.5	inf. precision				3.93	3.92
	12	12	8	8	4.31	4.53
	12	12	4	4	4.27	4.53
	12	16	8	8	3.94	3.93
0.25	inf. precision				3.35	3.35
	12	12	8	8	3.66	4.38
	12	12	4	4	3.64	4.38
	12	16	8	8	3.37	3.36
0.125	inf. precision				3.12	3.12
	12	12	8	8	3.46	5.04
	12	12	4	4	3.42	5.04
	12	16	8	8	3.13	3.14

where $\sigma_{q2}^2 = 2^{-2b_2}/12$ and $\sigma_{q1}^2 = N(2^{-2b_1} + 2^{-4B_d} - 2^{-b_1 - 2B_d + 1})/12$. For $2B_d$ much greater than b_1, $\sigma_{q1}^2 \approx N\ 2^{-2b_1}/12$. The value of b_3 is dependent on the length of the filter. It should have enough bits to represent the norm without saturating.

4 Simulation Results

A Gaussian sequence of zero mean and variance $\sigma_x^2 = 1/4$ was generated and used as input data, so that $\mathbf{R} = \sigma_x^2 \mathbf{I}$. The System Identification (SI) configuration [4] was used for all simulations. The system to be identified is an FIR filter with coefficients shown in Table I. This is a Prolate window with $\theta = 0.2\pi$ and order = 20. The adaptive filter's order is 19, so that ξ_{min} is given by the square of the last coefficient of the window times the variance of the signal ($\xi_{min} = w_{19}^{*2}\sigma_x^2$).

TABLE III. Values of ξ ($\times 10^{-6}$), theoretical and simulated, for LMS algorithm ($\mu = 0.108$).

B_d	B_c	Simul.	Theor. [3]
inf. precision		3.93	3.94
12	12	4.28	4.53
12	16	3.95	3.95

The first example checks the validity of the expressions obtained. Simulations were performed for several values of μ and different numbers of bits for B_d, B_c, b_1 and b_2, $c = 1$ (we only quantize the result of vector-vector multiplication), run for 3000 iterations, ensemble averaged over 100 independent trials. It can be seen from the simulation results in Table II that the use of the *Averaging Principle* doesn't affect accuracy very much. Note that all infinite precision values match closely. It is shown that the calculated values match the simulation and that the number of bits used for b_1 and b_2 produce little effect on the final total output error.

In order to compare the performance of LMS and NLMS, we first chose μ's that gave the same performance for both algorithms. The theoretical and simulated MSE for LMS algorithm are shown in Table III. The μ_{LMS} chosen here gives the same performance as $\mu_{NLMS} = 0.5$, as can be seen from the ξ using infinite precision in Table II. This shows that for this particular configuration, LMS and NLMS have almost the same performance, even under finite word-length effects.

Next, the effect of quantization on the updating of the coefficients is investigated. It's known that if some component of the product $\mu \mathbf{x}'_n e'_n$ is less in magnitude than 2^{-B_c-1}, the corresponding coefficient remains unchanged [3]. The condition for the adaptation to stop can be found in [3, 7] for LMS. For the NLMS we have

$$\mu^2 x_{rms}^2 E[f(\mathbf{x}_n)^2] E[e_n^2] < \frac{2^{-2B_c}}{4} \qquad (29)$$

where $x_{rms}^2 = \sigma_x^2$ and $E[e_n^2] = a^2\xi$. Table IV illustrates this effect. For μ satisfying (29), the total MSE is not well predicted by (25), as shown in Table IV.

Finally we analyze the effects of the division operation. When fewer bits are used for b_1, (28) shows that $E[f(\mathbf{x}_n)^2]$ decreases in value. From (25) we can see that decreasing $E[f(\mathbf{x}_n)^2]$ will also tend to decrease the total MSE. This means that a small look-up table can even improve slightly the performance of the algorithm. But if b_1 is too small, the quantization can no longer be modeled as additive noise and the above reasoning cannot be applied. Some simulation results are shown in Table II for $b_1 = b_2 = 4$ and 8 bits, and also in Table V.

TABLE IV. Values of ξ ($\times 10^{-6}$), theoretical and simulated, for μ that satisfies (29).

μ	B_d	B_c	b_1	b_2	Simul.	Theor.
1	10	10	8	8	8.96	13.44
0.5	10	10	8	8	5.69	13.62
0.25	10	10	8	8	6.40	19.85
0.125	10	10	8	8	29.45	33.82

TABLE V. Values of ξ ($\times 10^{-6}$), theoretical and simulated, for different values of b_1.

μ	B_d	B_c	b_1	b_2	Simul.	Theor.
0.25	12	12	6	8	3.645	4.380343
	12	12	4	8	3.642	4.380341
0.125	12	12	6	8	3.448	5.042042
	12	12	4	8	3.416	5.042040

5 Conclusion

A roundoff error analysis of the Normalized LMS algorithm has been presented. A closed-form expression for the total MSE in steady-state has been derived. It was shown that the theoretical results are in accordance with the experimental results, when the stalling phenomenon is not predominant. Implementation issues were considered. In particular, for the look-up table implementation of the division operation, it was shown that, in general, quantizing the norm will not deteriorate the performance of the algorithm. Simulations showing the amount of discrepancy due to approximations made are also provided.

Appendix A

In order to derive the mean-squared error due to NLMS, the *Averaging Principle* [5] was used. Using this approximation we can decouple $E[f(\mathbf{x}_n)\mathbf{x}_n\mathbf{x}_n^T]$ as $E[f(\mathbf{x}_n)]\mathbf{R}$. This approach has been used, for instance, in [5, 8], and it is a good approximation for Gaussian uncorrelated data (WGN), and/or for large N (order of the filter). Fig. 3 shows the discrepancy of this approximation. The relative error of the norm of the difference due to the approximation made is plotted for various values of N. Here we define the relative error as

$$\textit{Rel. error} = \frac{\left\| E\left[\frac{\mathbf{x}_n\mathbf{x}_n^T}{\mathbf{x}_n^T\mathbf{x}_n}\right] - \frac{E[\mathbf{x}_n\mathbf{x}_n^T]}{E[\mathbf{x}_n^T\mathbf{x}_n]} \right\|}{\left\| E\left[\frac{\mathbf{x}_n\mathbf{x}_n^T}{\mathbf{x}_n^T\mathbf{x}_n}\right] \right\|}. \qquad (30)$$

It is shown that for an eigenvalue spread ($\chi(\mathbf{R})$) equal to 1, the error due to the use of the *Averaging Principle* is at most 5%, while for high $\chi(\mathbf{R})$, but also large N, the error is still less than 10%.

It's easy to see that, using this approximation, the decoupled recursive expression of the correlation ma-

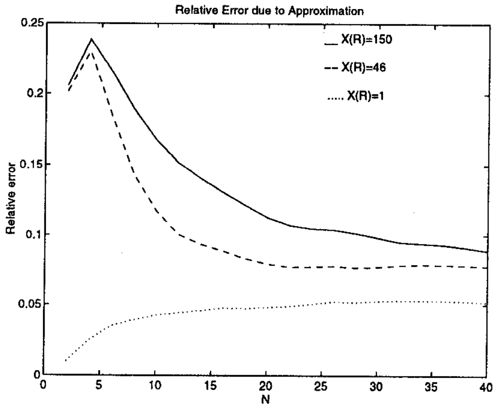

Fig. 3. Discrepancy due to approximation using Averaging Principle.

trix of the weight-error vector, as shown in [4] for LMS, will change to the following expression for NLMS

$$x^i_{(n+1)} = x^i_n - 2\mu\frac{\lambda_i x^i_n}{tr\mathbf{R}} + \mu^2\frac{\lambda_i \sum_{i=1}^N \lambda_i x^i_n}{(tr\mathbf{R})^2} + \mu^2\frac{\lambda_i^2 x^i_n}{(tr\mathbf{R})^2} + \mu^2\frac{\xi_{min}\lambda_i}{(tr\mathbf{R})^2} \quad (31)$$

where x^i_n is a diagonal element of $\mathbf{Q}^T\mathbf{K}_n\mathbf{Q}$, $\mathbf{K}_n = E[\epsilon_n\epsilon_n^T]$, $\epsilon_n = \mathbf{w}^* - \mathbf{w}_n$ and $\mathbf{Q}$ is an orthonormal transformation matrix. Consider the mean-squared error due to the NLMS algorithm, defined as [4]

$$\xi_n = E[e_n^2] = \xi_{min} + tr[\mathbf{R}\mathbf{K}_n] = \xi_{min} + \sum_{i=1}^N \lambda_i x^i_n. \quad (32)$$

In the limit, when n approaches infinity, $x^i_{(n+1)}$ and x^i_n become equal. Solving (31) for $\sum_{i=1}^N \lambda_i x^i_\infty$ we get

$$tr[\mathbf{R}\mathbf{K}_\infty] = \frac{\xi_{min}\mu\sum_{i=1}^N \frac{\lambda_i}{2tr\mathbf{R}-\mu\lambda_i}}{1-\mu\sum_{i=1}^N \frac{\lambda_i}{2tr\mathbf{R}-\mu\lambda_i}} \quad (33)$$

and the mean-squared error, using (32) and (33), will result in (10).

Appendix B

Here we note that

$$E[(\mathbf{w}_n^T\boldsymbol{\alpha}_n)^2] = E[\mathbf{w}_n^T\mathbf{w}_n]\sigma_d^2$$

$$= (E[\mathbf{w}^{*T}\mathbf{w}^*] + E[\epsilon_n^T\epsilon_n] + 2E[\epsilon_n^T\mathbf{w}^*])\sigma_d^2 \quad (34)$$

$$= (\|\mathbf{w}^*\|^2 + tr\mathbf{K}_n)\sigma_d^2 \quad (35)$$

The third term in (34) is zero because ϵ_n is uncorrelated with $\mathbf{w}^*$. Since we are interested in the steady-state behavior, i.e., $tr\mathbf{K}_\infty$, this value can also be obtained from (31) where we can explicitly find x^i_∞ and therefore

$$\sum_{i=1}^N x^i_\infty = tr\mathbf{K}_\infty = \sum_{i=1}^N \frac{\mu(\xi_{min} + tr\mathbf{R}\mathbf{K})}{2tr\mathbf{R} + \mu\lambda_i} \quad (36)$$

but $\xi_{min} + tr\mathbf{R}\mathbf{K} = \xi_\infty$ and this leads to the expression in (12).

References

[1] N. J. Bershad, "Analysis of the normalized LMS algorithm with Gaussian inputs", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no. 4, pp. 793-806, Aug. 1986.

[2] M. Tarrab and A. Feuer, "Convergence and performance analysis of the normalized LMS algorithms with uncorrelated Gaussian data", *IEEE Trans. Inform. Theory*, vol. IT-34, no. 4, pp. 680-691, July 1988.

[3] C. Caraiscos and B. Liu, "A roundoff error analysis of the LMS adaptive algorithm", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 32, no. 1, pp. 34-41, Feb. 1984.

[4] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1991, 2nd. Ed.

[5] C. G. Samson and V. U. Reddy, "Fixed point error analysis of the normalized ladder algorithm", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 31, pp. 1177-1191, Oct. 1983.

[6] D. T. M. Slock, "On the convergence behavior of the LMS and the normalized LMS algorithms", *IEEE Trans. on Signal Processing*, vol. 41, no. 9, pp. 2811-2825, Sept. 1993.

[7] N. J. Bershad, "Nonlinear quantization effects in the LMS and Block LMS adaptive algorithms - A comparison," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1504-1512, Oct. 1989.

[8] P. S. R. Diniz, M. L. R. de Campos and A. Antoniou, "Analysis of LMS-Newton adaptive filtering algorithms with variable convergence factor," *IEEE Trans. Signal Processing*, vol. 43, pp. 617-627, Mar. 1995.