Gradient-Descent Adaptive Filtering Using Gradient Adaptive Step-Size

Sayed Pouria Talebi*, Hossein Darvishi* Stefan Werner*, and Pierluigi Salvo Rossi*†
*Department of Electronic Systems, Norwegian University of Science and Technology, NO-7491 Norway
†Department of Gas Technology, SINTEFF Energy Research, Norway
{pouria,hossein.darvishi,stefan.werner}@ntnu.no, salvorossi@ieee.org

2022 IEEE 12th Sensor Array and Multichannel Signal Processing Workshop (SAM) | 978-1-6654-0633-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/SAM53842.2022.9827710

***Abstract*—At the heart of most adaptive filtering techniques lies an iterative statistical optimisation process. These techniques typically depend on adaptation gains, which are scalar parameters that must reside within a region determined by the input signal statistics to achieve convergence. This manuscript revisits the paradigm of determining near-optimal adaptation gains in adaptive learning and filtering techniques. The adaptation gain is considered as a matrix that is learned from the relation between input signal and filtering error. The matrix formulation allows adequate degrees of freedom for near-optimal adaptation, while the learning procedure allows the adaption gain to be formulated even in cases where the statistics of the input signal are not precisely known.**

***Index Terms*—Adaptive filtering, statistical estimation, gradient descent, optimisation.**

I. Introduction

A concept well understood in mathematics circles, the introduction of the least mean square (LMS) adaptive filtering technique [1] planted iterative statistical optimisation techniques at the heart of filtering solutions [2,3]. In contrast to its priors, such as the Wiener filtering solution, that calculated fixed filtering weights based on assumptions on the signal statistics, the LMS learned its filtering weights from the signal using an efficient gradient decent technique. The prospect of learning from the signal of interest has found the LMS and its numerous derivatives wide-ranging applications, and has garnered a great deal of interest in iterative optimisation solutions from the signal processing and machine learning communities [3,4].

Despite their vantage points, adaptive filtering and learning techniques built on the gradient descent method rely on a scalar parameter referred to as the adaptation gain, which controls the performance of the learning process. The adaptation gain itself has an acceptable range in which it has to reside to guarantee convergence [3]. Selection of the adaptation gain within the acceptable range, produces a design trade-off. On one hand, if selected on the lower end of the acceptable range, the adaptation process will be accurate but slow. On the other hand, if the adaptation gain is set on the upper end of the acceptable range, a fast learning rate is made possible at the cost of accuracy. Thus, presenting a speed-accuracy dilemma [5].

The fundamental role of the adaptation gain has prompted concerted research thrusts to either format an optimal value or adjust the adaptation gain itself to the signal statistics and filtering/learning demands, a chronicle of which can be found in [5]. Most notable efforts in this direction are the normalised LMS (NLMS) [2,6,7] that regulates the adaptation gain using the norm of the input signal, the framework in [8] that draw a duality between the LMS and Kalman filter in order to derive an optimal adaptation gain, frameworks akin to [9] that use fractional-order norms of an error measure and/or input signal to regulate the adaptation process, and importantly, the frameworks in [10]–[13] that use gradient descent techniques to learn the adaptation gain itself. However, to this point, proposals in this regard are either reliant on signal statistics, which are rarely available in real-world scenarios, or reliant on time averages that reduce the elegant simplicity of the LMS. Perhaps, most importantly, is the issue of a scalar adaptation gain in of itself. Given the trend towards large-scale sensors arrays, the need to offer increased degrees of freedom on the adaptation gain to achieve an acceptable fit for the filtering weights has become a pressing issue.

This manuscripts revisits the problem of gradient-based adaptive filtering, where the iterative optimisation process is conducted using an adaptation gain matrix that offers the degrees of freedom necessary to fine-tune the learning process. The derived adaptive filter exploits the relation between the input signal and filtering error twice. Once to adapt filtering weights and once to adjust the adaptation gain matrix. The performance of the derived framework is analysed, setting convergence criteria and clarifying the effect of design parameters. Finally, performance of the derived filter is demonstrated using both synthetically generated signals and real-world recordings.

Mathematical Notations: Scalars, column vectors, and matrices are denoted respectively by lowercase, bold lowercase, and bold uppercase letters, while $\mathbf{I}$ denotes an identity matrix of appropriate size. The transpose, statistical expectation, and spectral radius operators are denoted by $(\cdot)^{\mathsf{T}}$, $\mathsf{E}\left\{\cdot\right\}$, and $\mathsf{p}\left\{\cdot\right\}$, with ∇_{χ} indicating the gradient operator with respect to χ, while $\mathsf{vec}\left\{\cdot\right\}$ transforms a matrix into a column vector. Finally, the Kronecker product is denoted by $\otimes$.

II. Dual Gradient-Descent Adaptation

A. Problem Formulation

In general the goal is to find weighting matrix $\mathbf{W}_{\text{opt}}$ that best relates an input vectors sequence, $\{\mathbf{x}_n, n = 1, 2, 3, \ldots\}$ to an

This work was founded partially by the Research Council of Norway.

observed output vector sequence $\{\mathbf{y}_n, n = 1, 2, 3, \ldots\}$. This task is performed through an iterative optimisation process so that we have

$$\mathbf{W}_{n+1} = \mathbf{W}_n + \mathbf{G}_n \boldsymbol{\epsilon}_n \mathbf{x}_n^{\mathsf{T}} \tag{1}$$

where $\mathbf{G}_n$ is an adaptation gain matrix and

$$\boldsymbol{\epsilon}_n = \mathbf{y}_n - \mathbf{W}_n \mathbf{x}_n \tag{2}$$

is the filtering error. The iterations of (1) is aimed at minimising the second-order norm of $\boldsymbol{\epsilon}_n$, and thus, ensuring $\mathbf{W}_n \to \mathbf{W}_{\text{opt}}$ as $n \to \infty$. In this setting, the main issue becomes that of selecting a suitable adaptation gain matrix. In what follows, a mechanism for learning the adaptation gain matrix is derived.

B. Learning the Adaptation Gain Matrix

The aim is to find an adaptation gain matrix that will result in the updated weight matrix, $\mathbf{W}_{n+1}$, that is the best fit relating $\mathbf{x}_n$ and $\mathbf{y}_n$. To this end, we consider the post update error expressed as

$$\tilde{\boldsymbol{\epsilon}}_n = \mathbf{y}_n - \mathbf{W}_{n+1} \mathbf{x}_n. \tag{3}$$

Substituting (1) into (3) yields

$$\tilde{\boldsymbol{\epsilon}}_n = \mathbf{y}_n - \mathbf{W}_n \mathbf{x}_n - \mathbf{G}_n \boldsymbol{\epsilon}_n \mathbf{x}_n^{\mathsf{T}} \mathbf{x}_n$$

where using (2) we have

$$\tilde{\boldsymbol{\epsilon}}_n = \left(\mathbf{I} - \mathbf{G}_n \mathbf{x}_n^{\mathsf{T}} \mathbf{x}_n\right) \boldsymbol{\epsilon}_n$$

which allows a cost function to be formulated as

$$\mathcal{J}_n = \tilde{\boldsymbol{\epsilon}}_n^{\mathsf{T}} \tilde{\boldsymbol{\epsilon}}_n = \boldsymbol{\epsilon}_n^{\mathsf{T}} \left(\mathbf{I} - \mathbf{G}_n \mathbf{x}_n^{\mathsf{T}} \mathbf{x}_n\right)^{\mathsf{T}} \left(\mathbf{I} - \mathbf{G}_n \mathbf{x}_n^{\mathsf{T}} \mathbf{x}_n\right) \boldsymbol{\epsilon}_n. \tag{4}$$

The cost function in (4) allows the gain matrix to be learned in a gradient descent manner so that we have

$$\begin{aligned} \mathbf{G}_{n+1} =& \mathbf{G}_n - \mu \nabla_{\mathbf{G}_n} \mathcal{J}_n \\ =& \mathbf{G}_n + \mu \left(\mathbf{I} - \mathbf{G}_n \mathbf{x}_n^{\mathsf{T}} \mathbf{x}_n\right) \left(\mathbf{x}_n^{\mathsf{T}} \mathbf{x}_n\right) \boldsymbol{\epsilon}_n \boldsymbol{\epsilon}_n^{\mathsf{T}} \\ =& \mathbf{G}_n \left(\mathbf{I} - \mu \left(\mathbf{x}_n^{\mathsf{T}} \mathbf{x}_n\right)^2 \boldsymbol{\epsilon}_n \boldsymbol{\epsilon}_n^{\mathsf{T}}\right) + \mu \mathbf{x}_n^{\mathsf{T}} \mathbf{x}_n \boldsymbol{\epsilon}_n \boldsymbol{\epsilon}_n^{\mathsf{T}} \end{aligned} \tag{5}$$

where all constant terms have been incorporated into the real-valued positive adaptation gain μ.

The overall operation of the derived dual gradient descent technique is shown in Fig. 1, demonstrating the mechanism that allows the filtering error and input signal to be used twice. Once to adapt the filtering weight and once to learn the adaptation gain matrix. The performance of the derived framework is analysed in the sequel.

III. Performance Analysis and Convergence

Assume, without loss of generality, that $\mathbf{W}_1 = \mathbf{0}$ and $\mathbf{y}_n = \mathbf{W}_{\text{opt}} \mathbf{x}_n$. Then, from the expression in (2), we have

$$\boldsymbol{\epsilon}_1 = \mathbf{W}_{\text{opt}} \mathbf{x}_1$$

and therefore, (1) yields

$$\mathbf{W}_2 = \mathbf{G}_1 \mathbf{W}_{\text{opt}} \mathbf{x}_1 \mathbf{x}_1^{\mathsf{T}}.$$

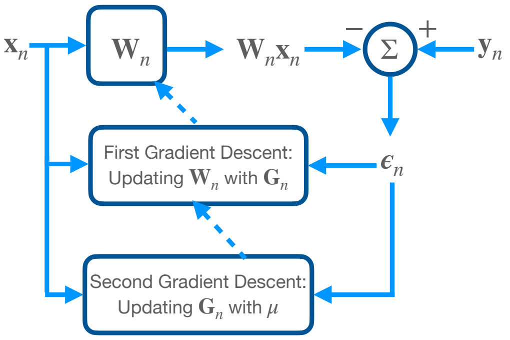
$\mathbf{x}_n$
$\mathbf{W}_n$
$\mathbf{W}_n\mathbf{x}_n$
$-$
Σ
$+$
$\mathbf{y}_n$
First Gradient Descent:
Updating $\mathbf{W}_n$ with $\mathbf{G}_n$
$\boldsymbol{\epsilon}_n$
Second Gradient Descent:
Updating $\mathbf{G}_n$ with μ

Fig. 1. Schematic showing the operations of the derived dual gradient descent mechanism.

Repeating this procedure n times, allows the weight matrix at time instant n to be expressed as

$$\begin{aligned} \mathbf{W}_n =& \sum_{i=1}^{n-1} \mathbf{G}_i \mathbf{W}_{opt} \mathbf{x}_i \mathbf{x}_i^{\mathsf{T}} \\ & - \sum_{j=2}^{n-1} \sum_{i=1}^{j-1} \mathbf{G}_j \mathbf{G}_i \mathbf{W}_{opt} \mathbf{x}_i \mathbf{x}_i^{\mathsf{T}} \mathbf{x}_j \mathbf{x}_j^{\mathsf{T}} \\ & + \sum_{k=3}^{n-1} \sum_{j=2}^{k-1} \sum_{i=1}^{j-1} \mathbf{G}_k \mathbf{G}_j \mathbf{G}_i \mathbf{W}_{opt} \mathbf{x}_i \mathbf{x}_i^{\mathsf{T}} \mathbf{x}_j \mathbf{x}_j^{\mathsf{T}} \mathbf{x}_k \mathbf{x}_k^{\mathsf{T}} - \cdots \end{aligned} \tag{6}$$

The expression in (6) can be reformulated in a more elegant manner as

$$\mathsf{vec}\left\{\mathbf{W}_n\right\} = \boldsymbol{\Gamma}_n \mathsf{vec}\left\{\mathbf{W}_{\text{opt}}\right\} \tag{7}$$

where

$$\begin{aligned} \boldsymbol{\Gamma}_n =& \sum_{i=1}^{n-1} \left(\mathbf{x}_i \mathbf{x}_i^{\mathsf{T}}\right) \otimes \mathbf{G}_i - \sum_{j=2}^{n-1} \sum_{i=1}^{j-1} \left(\mathbf{x}_j \mathbf{x}_j^{\mathsf{T}} \mathbf{x}_i \mathbf{x}_i^{\mathsf{T}}\right) \otimes \left(\mathbf{G}_j \mathbf{G}_i\right) \\ & + \sum_{k=3}^{n-1} \sum_{j=2}^{k-1} \sum_{i=1}^{j-1} \left(\mathbf{x}_k \mathbf{x}_k^{\mathsf{T}} \mathbf{x}_j \mathbf{x}_j^{\mathsf{T}} \mathbf{x}_i \mathbf{x}_i^{\mathsf{T}}\right) \otimes \left(\mathbf{G}_k \mathbf{G}_j \mathbf{G}_i\right) - \cdots . \end{aligned} \tag{8}$$

Now consider the deviation of weight matrix $\mathbf{W}_n$ from its optimal value $\mathbf{W}_{\text{opt}}$, given by

$$\boldsymbol{\mathcal{E}}_n = \mathsf{vec}\left\{\mathbf{W}_{\text{opt}}\right\} - \mathsf{vec}\left\{\mathbf{W}_n\right\}. \tag{9}$$

Substituting (7) into (9) gives

$$\boldsymbol{\mathcal{E}}_n = \left(\mathbf{I} - \boldsymbol{\Gamma}_n\right) \mathsf{vec}\left\{\mathbf{W}_{\text{opt}}\right\} \tag{10}$$

Then, from (10) and (8) it follows that for $\boldsymbol{\mathcal{E}}_n$, and by extension $\boldsymbol{\epsilon}_n$, to be exponentially bounded in the mean square sense, it is required that $\mathbf{G}_n$ be positive definite with

$$\mathsf{p}\left\{\mathbf{G}_n\right\} < \frac{1}{\mathsf{E}\left\{\mathbf{x}_n^{\mathsf{T}} \mathbf{x}_n\right\}} \tag{11}$$

which highlights the effect of $\mathbf{G}_n$ on filtering performance. Thus, the role and evolution of $\mathbf{G}_n$ including a number of special cases will be investigated in the next.

Remark 1. From (11), the safeguard step

$$\text{if } \mathsf{p}\left\{\mathbf{G}_n\right\} > \frac{1}{\mathsf{E}\left\{\mathbf{x}_n^\mathsf{T}\mathbf{x}_n\right\}} \quad \text{then } \mathbf{G}_n \leftarrow \mathbf{G}_n \frac{1}{\mathbf{x}_n^\mathsf{T}\mathbf{x}_n \mathsf{p}\left\{\mathbf{G}_n\right\}}$$

to ensure (11) is satisfied can be incorporated. Moreover, if the second-order statistic of $\mathbf{x}_n$ is not available, $\|\mathbf{x}_n\|^2$ can replace $\mathsf{E}\left\{\mathbf{x}_n^\mathsf{T}\mathbf{x}_n\right\}$.

In order to present an initial guideline and considering evolution of the adaptation gain matrix sequence, as given in (5), the adaptation gain is considered within the range of

$$0 < \mu < \frac{1}{\left(\mathbf{x}_n^\mathsf{T}\mathbf{x}_n\right)^2 \mathsf{p}\left\{\boldsymbol{\epsilon}_n\boldsymbol{\epsilon}_n^\mathsf{T}\right\}}. \tag{12}$$

Needless to state that for the lower end of (12), that is, as $\mu \to 0$, the adaptation gain matrix, $\mathbf{G}_n$ becomes time invariant and the algorithm operates akin to the LMS, where using an adaptation gain matrix offers the degrees of freedom necessary in array processing applications to tailor the filtering performance in a more accurate manner than the classical LMS. On the other hand, as

$$\mu \to \frac{1}{\left(\mathbf{x}_n^\mathsf{T}\mathbf{x}_n\right)^2 \mathsf{p}\left\{\boldsymbol{\epsilon}_n\boldsymbol{\epsilon}_n^\mathsf{T}\right\}}$$

the adaptation gain matrix evolutions in (5) has a stable point

$$\mathbf{G}_n \approx \frac{1}{\mathbf{x}_n^\mathsf{T}\mathbf{x}_n}\mathbf{I}.$$

Thus, making the filtering operation akin to that of the NLMS.

Remark 2. Note that, in most adaptive filtering scenarios, it would be prudent to assume $\|\boldsymbol{\epsilon}_n\|^2 << \|\mathbf{x}_n\|^2$. Therefore, $\mu = \frac{1}{(\mathbf{x}_n^\mathsf{T}\mathbf{x}_n)^2}$ falls within the range given in (11) and presents a normalised adaptation process for the adaptation gain matrix.

Finally, in order to present an overall perspective on the filtering operations, (5) is substituted into (1) to give

$$\begin{aligned}\mathbf{W}_{n+1} =& \mathbf{W}_n + \mathbf{G}_{n-1}\left(\mathbf{I} - \mu\left(\mathbf{x}_{n-1}^\mathsf{T}\mathbf{x}_{n-1}\right)^2 \boldsymbol{\epsilon}_{n-1}\boldsymbol{\epsilon}_{n-1}^\mathsf{T}\right)\boldsymbol{\epsilon}_n\mathbf{x}_n^\mathsf{T}\\ &+ \mu\mathbf{x}_{n-1}^\mathsf{T}\mathbf{x}_{n-1}\boldsymbol{\epsilon}_{n-1}\boldsymbol{\epsilon}_{n-1}^\mathsf{T}\boldsymbol{\epsilon}_n\mathbf{x}_n^\mathsf{T}\end{aligned}$$

which after some mathematical manipulation yields

$$\begin{aligned}\mathbf{W}_{n+1} =& \mathbf{W}_n + \mathbf{G}_{n-1}\boldsymbol{\epsilon}_n\mathbf{x}_n^\mathsf{T}\\ &- \mu\mathbf{G}_{n-1}\left(\mathbf{x}_{n-1}^\mathsf{T}\mathbf{x}_{n-1}\right)^2 \boldsymbol{\epsilon}_{n-1}\boldsymbol{\epsilon}_{n-1}^\mathsf{T}\boldsymbol{\epsilon}_n\mathbf{x}_n^\mathsf{T}\\ &+ \mu\mathbf{x}_{n-1}^\mathsf{T}\mathbf{x}_{n-1}\boldsymbol{\epsilon}_{n-1}\boldsymbol{\epsilon}_{n-1}^\mathsf{T}\boldsymbol{\epsilon}_n\mathbf{x}_n^\mathsf{T}.\end{aligned} \tag{13}$$

Now, assuming that the filtering iterations have converged to the point that $\mathbf{G}_n \approx \mathbf{G}_{n-1}$ and $\boldsymbol{\epsilon}_n \approx \boldsymbol{\epsilon}_{n-1}$ are reasonable approximations, from (13), we have

$$\begin{aligned}\mathbf{W}_{n+1} \approx& \mathbf{W}_n + \mathbf{G}_n\boldsymbol{\epsilon}_n\mathbf{x}_n^\mathsf{T}\\ &- \mu\mathbf{G}_n\left(\mathbf{x}_{n-1}^\mathsf{T}\mathbf{x}_{n-1}\right)^2\|\boldsymbol{\epsilon}_n\|^2\boldsymbol{\epsilon}_n\mathbf{x}_n^\mathsf{T}\\ &+ \mu\mathbf{x}_{n-1}^\mathsf{T}\mathbf{x}_{n-1}\|\boldsymbol{\epsilon}_n\|^2\boldsymbol{\epsilon}_n\mathbf{x}_n^\mathsf{T}\\ =& \mathbf{W}_n + \mathbf{G}_n\boldsymbol{\epsilon}_n\mathbf{x}_n^\mathsf{T} \qquad (14)\\ &+ \underbrace{\mu\left(\mathbf{x}_{n-1}^\mathsf{T}\mathbf{x}_{n-1}\mathbf{I} - \mathbf{G}_n\left(\mathbf{x}_{n-1}^\mathsf{T}\mathbf{x}_{n-1}\right)^2\right)}_{\mathbf{M}_n}\|\boldsymbol{\epsilon}_n\|^2\boldsymbol{\epsilon}_n\mathbf{x}_n^\mathsf{T}.\end{aligned}$$

The expression in (14) can be reformulated as

$$\begin{aligned}\mathbf{W}_{n+1} =& \mathbf{W}_n - \mathbf{G}_n\nabla_{\mathbf{W}_n}\left(\|\boldsymbol{\epsilon}_n\|^2\right)\\ &- \mathbf{M}_n\nabla_{\mathbf{W}_n}\left(\|\boldsymbol{\epsilon}_n\|^4\right).\end{aligned} \tag{15}$$

From (15), notice that the update to filtering weights $\mathbf{W}_n$ consists of two terms. The first term, $\mathbf{G}_n\boldsymbol{\epsilon}_n\mathbf{x}_n$, is the matrix LMS update minimising the second-order error measure, $\|\boldsymbol{\epsilon}_n\|^2$, using the matrix gain $\mathbf{G}_n$. The second term, $\mathbf{M}_n\|\boldsymbol{\epsilon}_n\|^2\boldsymbol{\epsilon}_n\mathbf{x}_n^\mathsf{T}$, that is, in its essence, minimising the forth-order error measure $\|\boldsymbol{\epsilon}_n\|^4$. Given this mixture and the behaviour of gradient-based adaptation techniques using higher-order measures of the filtering error, the derived algorithm is expected to achieve higher convergence rates than the of the LMS. This is shown in the next section using simulation examples.

IV. Numerical Examples

A. *Test Signal*

In the first set of simulations, a signal was generated, where $\mathbf{W}_{\text{opt}}$ was a randomly selected 6×4 matrix and the input signal $\mathbf{x}_n$ was a white Gaussian process. The LMS and the derived filtering technique were used to estimate the output sequence. Fig. 2, shows the mean square error (MSE) performance of the LMS, the derived filtering technique, i.e., adaptive gain matrix LMS, and the derived filtering technique with $\mu = 1/(\mathbf{x}_n^\mathsf{T}\mathbf{x}_n)$, i.e., normalised adaptive gain matrix LMS. The adaptation gains were set so that all filtering techniques achieved the same steady-state MSE, allowing the convergence rates to be compared. From Fig. 2, note that the derived filtering techniques converged and achieve faster convergence rates than that of the LMS.

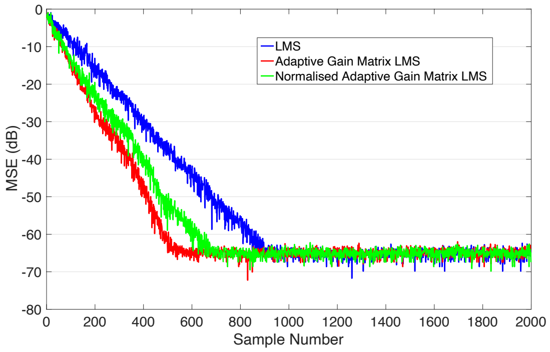

Fig. 2. MSE performance of the LMS and derived filtering techniques using synthetically generated signal, with adaptation gains set so that all filtering techniques achieve similar steady-state MSE.

In the second set of simulations, we had $\mathbf{G}_1 = \eta\mathbf{I}$ where η indicates the adaptation gain of the traditional LMS algorithm. In order to demonstrate the ability of the derived filtering techniques to learn from the input signal, η was selected to result in divergent filtering behaviour. Fig. 3, shows the MSE performance of the LMS and the adaptive gain matrix LMS. Observe that, despite the initial value of the adaptation gain matrices falling outside the convergence criteria, the derived techniques were able to adapt the gain matrix values resulting in convergent filtering behaviour, demonstrating their ability to learn these values form the data itself.

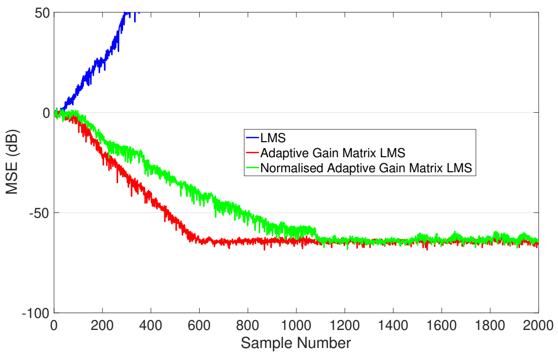

Fig. 3. MSE performance of the LMS and derived filtering techniques using synthetically generated signal, with initial adaptation gain matrices set to $\mathbf{G}_1 = \eta\mathbf{I}$, where η indicates the adaptation gain of the LMS filter.

B. *Real-World Recording*

The data set used in this set of simulations was collected at the University of North Carolina at Greensboro [14]. The data set was obtained from a single-hop and a multi-hop wireless sensor network using TelosB motes and contains four sensors located indoor and outdoor, recording humidity and temperature. Recordings were collected for 6 hours at 5 seconds intervals. Labelled anomalies injected into the data-set were discarded, and only the temperature recordings of a single indoor and a single outdoor sensor from the multi-hop section of the data set were considered in this simulation.[1]

The derived filtering techniques were used to predict temperatures in the next two time steps using the data of the past six observations (i.e., a sliding window). The performance of the derived filtering techniques is benchmarked against that of the LMS in Fig. 4. Note that the derived technique with adaptation of the gain matrix converged and followed statistical changes of the data in an agile manner, while achieving a comparable steady-state MSE to that of the LMS.

Finally, the same simulation was carried out on data recorded using outdoor sensors. The performance of the derived filtering technique is shown in Fig. 5. Observe that once more, the derived techniques achieved similar steady-state MSE performance to that of the LMS while exhibiting faster initial convergence and response to statistical changes in the data.

V. Conclusion

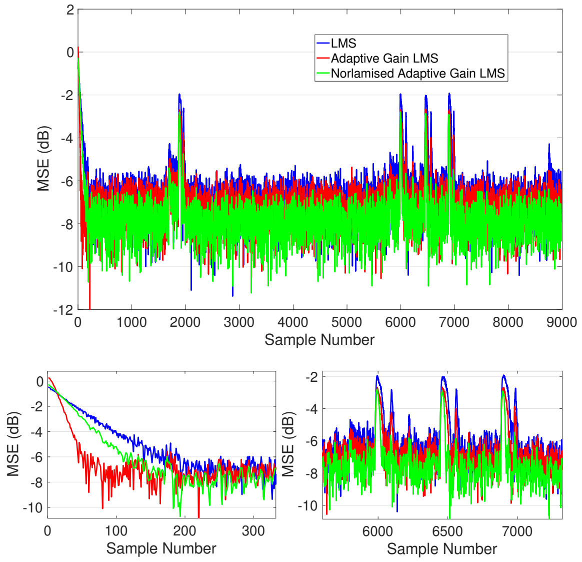

Fig. 4. Temperature prediction performance using indoor sensor recordings. Top graph displays the full simulation result while performance during initial stages and change of underlying statistics is shown in the bottom graphs.

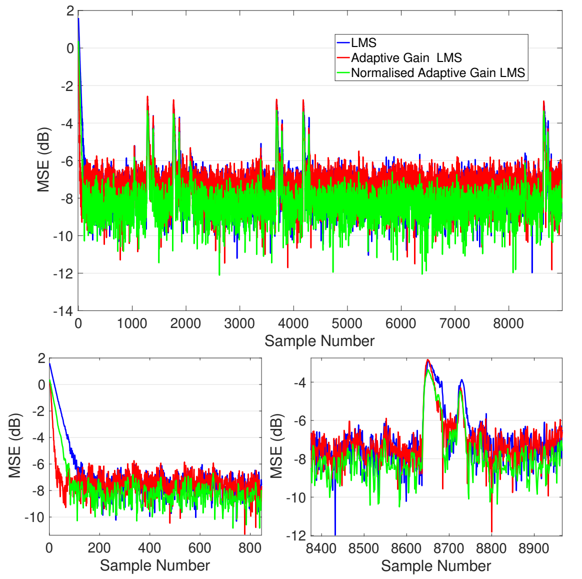

Fig. 5. Temperature prediction performance using outdoor sensor recordings. Top graph displays the full simulation result while performance during initial stages and change of underlying statistics is shown in the bottom graphs.

The concept of iterative optimisation techniques used in adaptive filtering has been revisited and an adaptive gradient-descent technique for adaptive array processing has been derived. The use of a matrix adaptation gain, which is learned from the input signal, allows for processing of high-dimensional signals encountered in array signal processing. The derived filtering concept has shown promise in simulations using real-world data and the operations of the derived filtering solution has been analysed, indicating that more elaborate solutions await derivation. Thus, opening a new area of research.

[1] The WSN data set is publicly available and can be found online at https://home.uncg.edu/cmp/downloads/lwsndr.html

REFERENCES

[1] B. Widrow and M. E. Hoff, "Adaptive switching circuits," *IRE WESCON Convention Record, Part 4*, pp. 96–104, August 1960.

[2] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Prentice Hall, 1985.

[3] D. P. Mandic and J. A. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*. John Wiley & Sons, 2001.

[4] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Prentice Hall, 2000.

[5] D. P. Mandic and V. S. L. Goh, *Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear, and Neural Models*. Wiley, 2009.

[6] A. E. ALBERT and L. A. Gardne, *Stochastic Approximation and Nonlinear Regression*. MIT Press, 1967.

[7] N. J. Bershad and J. C. Bermudez, "A switched variable step size NLMS adaptive filter," *Digital Signal Processing*, vol. 101, p. 102730, 2020.

[8] D. P. Mandic, S. Kanna, and A. G. Constantinides, "On the intrinsic relationship between the least mean square and Kalman filters [Lecture Notes]," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 117–122, 2015.

[9] S. P. Talebi, S. Werner, S. Li, and D. P. Mandic, "Tracking dynamic systems in α-stable environments," *In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4853–4857, 2019.

[10] V. J. Mathews and Z. Xie, "A stochastic gradient adaptive filter with gradient adaptive step size," *IEEE Transactions on Signal Processing*, vol. 41, no. 6, pp. 2075–2087, 1993.

[11] W. P. Ang and B. Farhang-Boroujeny, "A new class of gradient adaptive step-size LMS algorithms," *IEEE Transactions on Signal Processing*, vol. 49, no. 4, pp. 805–810, 2001.

[12] E. C. Mengüç and N. Acir, "An augmented complex-valued least-mean kurtosis algorithm for the filtering of noncircular signals," *IEEE Transactions on Signal Processing*, vol. 66, no. 2, pp. 438–448, 2018.

[13] P. A. C. Lopes, "Bayesian step least mean squares algorithm for Gaussian signals," *IET Signal Processing*, vol. 14, no. 2, pp. 506–512, September 2020.

[14] S. Suthaharan, M. Alzahrani, S. Rajasegarar, C. Leckie, and M. Palaniswami, "Labelled data collection for anomaly detection in wireless sensor networks," *In Proceedings of IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 269–274, 2010.