A FULLY ADAPTIVE NORMALIZED NONLINEAR GRADIENT DESCENT ALGORITHM FOR NONLINEAR SYSTEM IDENTIFICATION

Igor R. Krcmar

Faculty of Electrical Engineering
University of Banjaluka
Banjaluka, BH.
ikrcmar@etf-bl.rstel.net

Danilo P. Mandic

School of Information Systems
University of East Anglia
Norwich, UK.
d.mandic@sys.uea.ac.uk

ABSTRACT

A fully adaptive normalized nonlinear gradient descent (FANNGD) algorithm for neural adaptive filters employed for nonlinear system identification is proposed. This full adaptation is achieved using the instantaneous squared prediction error to adapt the free parameter of the NNGD algorithm. The convergence analysis of the proposed algorithm is undertaken using contractivity property of the nonlinear activation function of a neuron. Simulation results show that a fully adaptive NNGD algorithm outperforms the standard NNGD algorithm for nonlinear system identification.

1. INTRODUCTION

A single-neuron nonlinear adaptive filter, trained by gradient descent (GD) can be described by [1]

$$e(k) = d(k) - \Phi(\mathbf{x}^T(k)\mathbf{w}(k)) \tag{1}$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \eta\nabla_{\mathbf{w}}E(e(k)) \tag{2}$$

where k denotes a discrete time instant, $e(k)$ is the instantaneous error at the neuron, $E(\cdot)$ is the filter cost function, $d(k)$ is some teaching (desired) signal, $\mathbf{x}(k) = [x_1(k), ..., x_N(k)]^T$ is the input vector, $\mathbf{w}(k) = [w_1(k), ..., w_N(k)]^T$ is the weight vector, $\Phi(\cdot)$ denotes a nonlinear activation function of a neuron, η denotes the learning rate and $(\cdot)^T$ denotes the vector transpose. The most common choice for the cost function $E(\cdot)$ is

$$E(e(k)) = \frac{1}{2}e^2(k) \tag{3}$$

The algorithm for on-line adaptation of such an adaptive filter, described by the equations (1), (2), and (3), is referred to as the Nonlinear Gradient Descent algorithm (NGD) [1]. Due to its simplicity and inherent nonlinearity, this filter might be considered as a suitable choice for applications in nonlinear and/or nonstationary system identification.

The NGD algorithm, however, might suffer from slow convergence and local minima on the error performance surface of the filter. A fixed learning rate parameter η can be considered as one of the factors contributing to these problems. Further, a recent result [2] indicates an inherent relationship between the learning rate parameter η and steepness of the nonlinear activation function of β, which has a negative impact on the convergence properties of the NGD algorithm with fixed η. In order to achieve fast convergence in the beginning of adaptation of a filter trained by the NGD algorithm, it is advisable to adopt a large learning rate parameter η. On the other hand, to ensure convergence to an optimal weight vector $\mathbf{w}^*$, when close to the optimal solution the learning rate parameter has to be small [3]. Also, a large η may result in instability of the algorithm. The cost function given in (3) represents an instantaneous estimate of the ensemble average $\langle e^2(k)\rangle$, thus introducing a gradient noise in the operation of the algorithm [4]. This noise will help the algorithm to escape from spurious local minima, but will also reduce the convergence rate of the algorithm.

For nonlinear systems, learning algorithms with an adaptive learning rate are most desirable [5, 1]. One such algorithm for adaptation of a GD based, single-neuron neural adaptive filter is given in [1]. The Normalized Nonlinear Gradient Descent (NNGD) algorithm exhibits optimal behaviour in the sense that it minimizes the instantaneous prediction error, by means of an adaptive learning rate η.

Here we propose a fully adaptive NNGD (FANNGD) with an adaptive learning rate, which is based upon adaptation of the free parameter of the NNGD algorithm using the instantaneous squared prediction error. The convergence analysis of the proposed algorithm is carried out based upon the contraction mapping properties of the nonlinear activation function of a neuron. The analysis is supported by examples for nonlinear system identification and Monte Carlo analysis.

2. THE NNGD ALGORITHM

Following the method given in [1], when the error term (1), is expanded with a Taylor series, we have

$$e(k+1) = e(k) + \sum_{i=1}^{N}\frac{\partial e(k)}{\partial w_i(k)}\triangle w_i(k)$$
$$+\frac{1}{2!}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\partial^2 e(k)}{\partial w_i(k)\partial w_j(k)}\triangle w_i(k)\triangle w_j(k) + \cdots \tag{4}$$

From (1), the first partial derivative can be obtained as

$$\frac{\partial e(k)}{w_i(k)} = -\Phi'(net(k))x_i(k); \quad i = 1, 2, \ldots, N \tag{5}$$

where $net(k) = \mathbf{x}^T(k)\mathbf{w}(k)$. From (2) and (3), the weight update for $i = 1, 2, \ldots, N$ is given by

$$\begin{aligned}\Delta w_i(k) &= w_i(k+1) - w_i(k)\\ &= \eta\Phi'(net(k))e(k)x_i(k)\end{aligned} \tag{6}$$

The second partial derivatives for $i, j = 1, 2, \ldots, N$ can be calculated as

$$\frac{\partial^2 e(k)}{\partial w_i(k) \partial w_j(k)} = -\Phi''(net(k)) x_i(k) x_j(k) \qquad (7)$$

Combining equations (4), (5), (6), and (7), yields

$$\begin{aligned} e(k+1) &= e(k) - \eta[\Phi'(net(k))]^2 e(k) \sum_{i=1}^{N} x_i^2(k) \\ &- \frac{1}{2!}\eta^2 e^2(k)[\Phi'(net(k))]^2 \\ &\times \Phi''(net(k)) \sum_{i=1}^{N}\sum_{j=1}^{N} x_i^2(k) x_j^2(k) + \cdots \quad (8) \end{aligned}$$

If, for simplicity, we neglect the second and higher order derivatives of Φ, then the error term in (8) becomes

$$e(k+1) = e(k)\Big[1 - \eta[\Phi'(net(k))]^2 \|\mathbf{x}(k)\|_2^2\Big] \qquad (9)$$

where $\| \cdot \|_2$ denotes the Euclidean norm. In equation (9), $e(k+1)$ equals zero for an optimal learning rate

$$\eta_{OPT}(k) = \frac{1}{[\Phi'(net(k))]^2 \|\mathbf{x}(k)\|_2^2} \qquad (10)$$

The contribution of the second and higher order terms from (8) to the learning rate given in (10) can be compensated by adding a variable C to the denominator of (10). Thus an optimal learning rate for the NNGD algorithm becomes

$$\eta_{OPT}(k) = \frac{1}{C + [\Phi'(net(k))]^2 \|\mathbf{x}(k)\|_2^2} \qquad (11)$$

The physical meaning of such an adaptive learning rate is the self-normalization of the algorithm, since the magnitude of the learning rate varies in time depending on the tap input power and gradient in the state space of the filter. The learning rate of the NNGD algorithm is not fully adaptive since C was chosen to be constant, and as a result, the estimation error is never zero. Usually, C is chosen as a small positive value, as in the Normalized Least Mean Squares (NLMS) algorithm, in order to ensure that the algorithm does not diverge [6]. To circumvent this drawback, we introduce a fully adaptive NNGD algorithm.

3. DERIVATION OF THE FANNGD ALGORITHM

Let us briefly analyse the Taylor series expansion given in (8). From equations (4), (5), (6), and (7), we can compute the reminder of the Taylor series expansion as

$$\begin{aligned} R_n(k) &= -\frac{1}{2!}\eta^2 e^2(k)[\Phi'(net(k))]^2 \\ &\times \Phi''(\mathbf{x}^T(k)(\mathbf{w}(k) + \theta\triangle\mathbf{w}(k)) \|\mathbf{x}(k)\|_2^4 \quad (12) \end{aligned}$$

where $0 < \theta \leq 1$. This reminder can be expressed as

$$R_n(k) = -\eta e(k) C^*(k) \qquad (13)$$

where

$$\begin{aligned} C^*(k) &= \frac{1}{2!}\eta e(k)[\Phi'(net(k))]^2 \\ &\times \Phi''(\mathbf{x}^T(k)(\mathbf{w}(k) + \theta\triangle\mathbf{w}(k)) \|\mathbf{x}(k)\|_2^4 \quad (14) \end{aligned}$$

An explicit computation of $C^*(k)$ from (14) would require to solve a quadratic equation, due to the relationship between the learning rate parameter η and constant C, given in (11). This is not easy, due to the fact that the discriminant of the quadratic equation can be negative. In addition, we can only estimate the value of the second derivative, which appears in the right hand side of (14).
Let us rewrite (8) as

$$\begin{aligned} e(k+1) = \big[1 &- \eta[\Phi'(net(k))]^2 \|\mathbf{x}(k)\|_2^2 \\ &- \frac{1}{2!}\eta^2 e(k)[\Phi'(net(k))]^2 \|\mathbf{x}(k)\|_2^2 \\ &\times \Phi''(\mathbf{x}^T(k)(\mathbf{w}(k) + \theta\triangle\mathbf{w}(k)) \|\mathbf{x}(k)\|_2^4\big] e(k) \quad (15) \end{aligned}$$

From (11) and (15) we have

$$e(k+1) = \gamma(k) e(k) \qquad (16)$$

where $\gamma(k) = \gamma[C(k)]$. Thus, the instantaneous squared error to be minimized is given by

$$e^2(k+1) = \gamma^2 e^2(k) \qquad (17)$$

This new function γ^2 is continuous on the set

$$\bar{\mathbf{R}} = \mathbf{R} \setminus \Big[-[\Phi'(net(k))]^2 \|\mathbf{x}(k)\|_2^2 \Big] \qquad (18)$$

where $\mathbf{R}$ denotes the set of real numbers. According to the Intermediate Value Theorem (IVT), function γ^2 attains its minimal value for some $C^*(k) \in \bar{\mathbf{R}}$, which does not have to be zero. From (15) and (16), it can be easily shown that $C^*(k)$ has a finite value, and that it is bounded from below by

$$\frac{-[\Phi'(net(k))]^2 \|\mathbf{x}(k)\|_2^2}{2} \leq C^*(k) \qquad (19)$$

Explicit computation of $C^*(k)$, that minimizes γ^2 at every discrete time instant k is computationally very expensive. Furthermore, it might result in large oscillations of the learning rate, and consequently in instability of the algorithm. Also, it does not ensure a safe start of the algorithm.
Therefore, in order to obtain a fully adaptive learning rate for the NNGD algorithm, we look for a sequence $C(k), k = 0, 1, 2, \ldots$, which minimizes (15). In addition, we would like to be able to obtain a sequence $C(k)$ from a simple recursive equation

$$C(k) = C(k-1) + \triangle C(k) \qquad (20)$$

which is not computationally expensive. Therefore, we propose computation of $C(k)$ according to the following recursive equation

$$C(k) = C(k-1) + \mu e^2(k) \qquad (21)$$

where μ is a small positive constant. To ensure a safe start of the algorithm, the initial condition $C(0)$ should be chosen as a small positive value. The proposed algorithm hence increases computational burden by two additional multiplications and one addition. Also, as desired, the sequence $C(k)$ computed according to (21) is monotonically increasing, thus providing a larger learning rate in the beginning of the operation of the algorithm (search phase), and reducing the value of the learning rate in the converge phase.

4. CONVERGENCE OF THE PROPOSED ALGORITHM

It is desirable that $|e(k)| \to 0$ as $k \to \infty$, which for equation (16) gives

$$|e(k+1)| = |\gamma||e(k)| \tag{22}$$

From (15), (16), and (22) we have

$$\begin{aligned} |e(k+1)| &= |1 - \eta[\Phi'(net(k))]^2 \|\mathbf{x}(k)\|_2^2 - \eta C^*(k)||e(k)| \\ &\leq |e(k)||1 - \eta[\Phi'(net(k))]^2 \|\mathbf{x}(k)\|_2^2 - \eta C^*(k)| \end{aligned} \tag{23}$$

From (23), $e(k)$ will converge *uniformly* if

$$|1 - \eta[\Phi'(net(k))]^2 \|\mathbf{x}(k)\|_2^2 - \eta C^*(k)| < 1 \tag{24}$$

which is a contractive behaviour. The last relationship can be rewritten as

$$-1 < 1 - \eta\big[[\Phi'(net(k))]^2 \|\mathbf{x}(k)\|_2^2 + C^*(k)\big] < 1 \tag{25}$$

Having in mind bounds on $C^*(k)$, the right hand side of inequality (25) can be easily verified. The left hand side of inequality (25) reduces to

$$[\Phi'(net(k))]^2 \|\mathbf{x}(k)\|_2^2 + 2C(k) > C^*(k) \tag{26}$$

Given that $C(k)$ has a monotonically increasing trend and $C^*(k)$ is finite, there is a discrete time instant K, so that for $k > K$ the inequality (26) holds. Due to a contractive behaviour of (22) and (23), it is obvious that in this case $e(k)$ exponentially decreases toward zero as $k \to \infty$, which implies $C(k) \to C(\infty)$ as $k \to \infty$, where $C(\infty)$ denotes some finite positive value.

5. MORE ON CONVERGENCE

Consider the error equation $e(k) = d(k) - \Phi(net(k))$, but assume

$$d(k) = q(k) + \Phi(\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)) \tag{27}$$

where $\tilde{\mathbf{w}}(k)$ are optimal filter weights and $q(k)$ denotes a zero mean uncorrelated measurement disturbance sequence, with variance σ_q^2. It then follows that

$$e(k) = q(k) + \Phi(\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)) - \Phi(net(k)) \tag{28}$$

Also, we shall assume a ”random walk” model for the optimal filter weights, i.e.

$$\tilde{\mathbf{w}}(k+1) = \tilde{\mathbf{w}}(k) + \varepsilon(k) \tag{29}$$

where $\varepsilon(k)$ is zero mean white vector process with covariance matrix $\sigma_\varepsilon^2\mathbf{I}$, where the $\mathbf{I}$ denotes the unitary matrix. Consider again the weight update equation

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \eta(k)e(k)\Phi'(net(k))\mathbf{x}(k) \tag{30}$$

From (28) and (30), we have

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) + \eta(k)q(k)\Phi'(net(k))\mathbf{x}(k) \\ &+\eta(k)[\Phi(\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)) - \Phi(net(k))]\Phi'(net(k))\mathbf{x}(k) \end{aligned} \tag{31}$$

Now, introduce the misalignment vector as

$$\mathbf{v}(k) = \tilde{\mathbf{w}}(k) - \mathbf{w}(k) \tag{32}$$

Following the approach from [7], if we subtract (31) from (29) yields

$$\begin{aligned} \mathbf{v}(k+1) &= \mathbf{v}(k) + \varepsilon(k) - \eta(k)q(k)\Phi'(net(k))\mathbf{x}(k) \\ &-\eta(k)[\Phi(\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)) - \Phi(net(k))]\Phi'(net(k))\mathbf{x}(k) \end{aligned} \tag{33}$$

For Φ a contraction mapping the term in the square brackets from (33) is bounded from above by $\alpha|\mathbf{x}^T(k)\mathbf{v}(k)|$, $0 < \alpha \leq 1$. Therefore,

$$\begin{aligned} \mathbf{v}(k+1) &\leq \mathbf{v}(k) + \varepsilon(k) - \eta(k)q(k)\Phi'(net(k))\mathbf{x}(k) \\ &- \eta(k)\mathbf{x}^T(k)\mathbf{v}(k)\alpha\Phi'(net(k))\mathbf{x}(k) \end{aligned} \tag{34}$$

For a contractive activation function of a neuron, term $\Phi'(net(k))$ is also bounded as $0 < \Phi'(net(k)) \leq 1$, and can be replaced by $\Phi'(\cdot) < \delta \leq 1$. Note that for a sigmoidal nonlinearity $\Phi'(\cdot) > 0$. Now (34) becomes

$$\begin{aligned} \mathbf{v}(k+1) &\leq \mathbf{v}(k) + \varepsilon(k) - \eta(k)q(k)\delta\mathbf{x}(k) \\ &- \eta(k)\mathbf{x}^T(k)\mathbf{v}(k)\alpha\delta\mathbf{x}(k) \end{aligned} \tag{35}$$

It is convenient to introduce the independence assumption between η, $\mathbf{x}$, and $\mathbf{v}$ which gives

$$E\big[\mathbf{v}(k+1)\big] = E\big[\mathbf{v}(k)\big]E\big[\mathbf{I} - \eta(k)\delta\mathbf{x}(k)\mathbf{x}^T(k)\alpha\big] \tag{36}$$

where $E(\cdot)$ is the expectation operator. For convergence, $0 < E\big[\|\mathbf{I} - \eta(k)\delta\mathbf{x}(k)\mathbf{x}^T(k)\alpha\|\big] < 1$, which for the upper limit of α and δ, and using an assumption that $x(k)$ is an uncorrelated input sequence gives

$$0 < \eta(k) < E\left[\frac{1}{\mathbf{x}^T(k)\mathbf{x}(k)}\right] \tag{37}$$

The learning rate of the proposed algorithm satisfies relationship (37). This means that the NLMS algorithm is the upper bound for the analyzed algorithm, which ensures stability of the proposed algorithm.
Finally, from (28) and using the assumption that Φ is a contraction mapping, gives

$$e^2(k) = q^2(k) + \alpha\mathbf{x}^T(k)\mathbf{v}(k)\mathbf{v}^T(k)\mathbf{x}(k) + 2\alpha q(k)\mathbf{x}^T(k)\mathbf{v}(k) \tag{38}$$

Thus, for the upper limit of α and taking expectation of both sides of (38), conditioned on the measurements taken up to the discrete time instant k, we have

$$\sigma_e^2 \leq \sigma_q^2 + \mathbf{x}^T(k)E\big[\mathbf{v}(k)\mathbf{v}^T(k)\big]\mathbf{x}(k) \tag{39}$$

where σ_e^2 denotes the variance of the prediction error. From (38) and (39), it is obvious that the sequence $C(k)$, computed according to equation (21), provides a time average of the estimate of the prediction error variance, which depends on the measurement noise variance and the covariance matrix of the misalignment vector.

6. EXPERIMENTAL RESULTS

In order to verify the analysis we performed several experiments on a nonlinear system described by

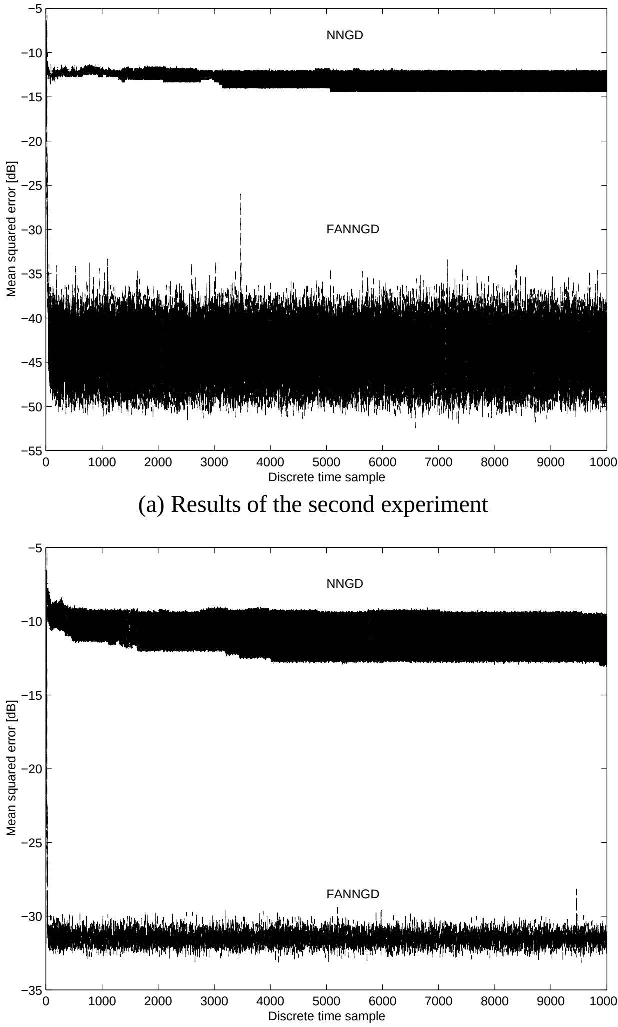
$$y(k) = \frac{1}{1 + e^{-\beta\mathbf{x}^T(k)(\mathbf{w}_0 + \triangle\mathbf{w}_0(k))}} + q(k) \tag{40}$$

where the weights $\mathbf{w}_0 = [\ -0.31962,\ 0.49231,\ -0.19933,\ 0.86133,\ -0.21727,\ -0.02523,\ -0.86119,\ -0.86022,\ -0.34348,\ 1.05471\ ]^T$, $\triangle \mathbf{w}_0(k)$ represents a zero mean uncorrelated vector process and $\beta = 4$. In all the experiments the filter order was $N = 10$ whereas the logistic function was employed within the neuron. The quantitative performance measure was the standard gain, a logarithmic ratio between the variances of the expected system output and the error $R_p = 10 \log(\hat{\sigma}_y^2/\hat{\sigma}_e^2)$. Constant C in the NNGD algorithm was set to be $C = 0.0001$, and parameter μ in the proposed algorithm was $\mu = 0.1$.

In the first experiment we compared the performances of the proposed algorithm and the standard NNGD algorithm for nonlinear system identification. Parameter $q(k)$ (39) was set to be $q(k) = 0, \forall k$, and $\triangle \mathbf{w}_0(k)$ was set to be $\triangle \mathbf{w}_0(k) = \mathbf{0}$. The input signal was a white noise sequence, appropriately scaled to fit the range of the logistic nonlinearity. In that case the standard gain for the fully adaptive NNGD algorithm was $R_p = 34.96dB$, while for the standard NNGD algorithm $R_p = 34.64dB$, which is more than $0.3dB$ better. In the second experiment we performed the Monte Carlo analysis of the proposed algorithm and the standard NNGD algorithm for identification of the nonlinear system (39), with $\triangle \mathbf{w}_0(k) \neq \mathbf{0}$ and $q(k) = 0, \forall k$. The number of iterations was 300. The results of the experiment are shown in Figure 1(a). In this case an average standard gain for the fully adaptive NNGD algorithm was $R_p = 31.77dB$, while for the standard NNGD algorithm $R_p = 19.71dB$.

In the third experiment we performed Monte Carlo analysis of the proposed algorithm and the standard NNGD algorithm for identification of nonlinear system (39), with $\triangle \mathbf{w}_0(k)$ set to be the same as in the second experiment and $q(k)$ was set to be a zero mean white process with variance $\sigma_q^2 = 0.01$. The number of iterations was 300. The results of the experiment are shown in Figure 1(b). In this case, an average gain for the fully adaptive NNGD algorithm was $R_p = 24.80dB$, while for the standard NNGD algorithm $R_p = 14.63dB$.

It is obvious that the proposed algorithm outperforms the standard NNGD algorithm for nonlinear system identification of system (40).



(a) Results of the second experiment

(b) Results of the third experiment

Fig. 1. Results of the Monte Carlo analysis

7. SUMMARY

A fully adaptive NNGD (FANNGD) algorithm for nonlinear system identification has been proposed. The proposed algorithm provides a full adaptation of the learning rate using an instantaneous squared prediction error, with a small increase in the computational burden, as compared with the NNGD algorithm. The convergence analysis of the proposed algorithm has been undertaken based on contractivity of the nonlinear activation function of a neuron. Experiments on nonlinear system identification support the analysis and confirm that the proposed algorithm outperforms the standard NNGD algorithm for system identification.

8. REFERENCES

[1] D. P. Mandic, ”NNGD algorithm for neural adaptive filters,” *Electronic Letters*, vol. 36, no. 9, pp. 845-846, 2000.

[2] D. P. Mandic and J. A. Chambers, ”Relationship between the slope of the activation function and the learning rate for the RNN,” *Neural Computation*, vol. 11, no. 5, pp. 1069-1077, 2000.

[3] 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.

[4] P. E. An, M. Brown, and C. J. Harris, ”Aspects of instantaneous on-line learning rules,” in *Proceedings of the Control ’94*, pp. 646-651, 1994.

[5] S. C. Douglas, ”A family of normalized LMS algorithms,” *IEEE Signal Processing Letters*, vol. 1, no. 3, pp. 49-51, 1994.

[6] S. Haykin, *Neural networks - A comprehensive foundation*, Prentice Hall, 1994.

[7] J. A. Chambers, W. Sherrliker, and D. P. Mandic, ”A normalized gradient algorithm for an adaptive recurrent perceptron,” in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP-2000)*, vol. 1, pp. 3530-3533, 2000.