EURASIP Journal on Applied Signal Processing 2003:12, 1229–1237

Nonlinear System Identification Using Neural Networks Trained with Natural Gradient Descent

Mohamed Ibnkahla

Electrical and Computer Engineering Department, Queen's University, Kingston, Ontario, Canada K7L 3N6
Email: mohamed.ibnkahla@ece.queensu.ca

Received 13 December 2002 and in revised form 17 May 2003

We use natural gradient (NG) learning neural networks (NNs) for modeling and identifying nonlinear systems with memory. The nonlinear system is comprised of a discrete-time linear filter H followed by a zero-memory nonlinearity $g(\cdot)$. The NN model is composed of a linear adaptive filter Q followed by a two-layer memoryless nonlinear NN. A Kalman filter-based technique and a search-and-converge method have been employed for the NG algorithm. It is shown that the NG descent learning significantly outperforms the ordinary gradient descent and the Levenberg-Marquardt (LM) procedure in terms of convergence speed and mean squared error (MSE) performance.

Keywords and phrases: satellite communications, system identification, adaptive signal processing, neural networks.

1. INTRODUCTION

Most techniques that have been proposed for nonlinear system identification are based on parametrized nonlinear models such as Wiener and Hammerstein models [1, 2, 3, 4], Volterra series [5], wavelet networks [3], neural networks (NNs) [6, 7], and so forth. The estimation of the parameters is performed either using nonadaptive techniques such as least squares methods and higher-order statistics-based methods [4, 8, 9, 10, 11], or adaptive techniques such as the backpropagation (BP) algorithm [12, 13, 14] and online learning [3, 15].

NN approaches for modeling and identifying nonlinear dynamical systems have shown excellent performance compared to classical techniques [1, 6, 9, 13, 16].

NNs trained with the BP algorithm [14, 16] have, however, two major drawbacks: first, their convergence is slow, which can be inadequate for online training; second, the NN parameters may be trapped in a nonoptimal local minimum, leading to suboptimal approximation of the system [6]. Natural gradient (NG) learning [17, 18] on the other hand, has been shown to have better convergence capabilities than the classical BP algorithm because it takes into account the geometry of the manifold in which the NN weights evolve. Therefore, NG learning can better avoid the plateau phenomena, which characterize the BP learning curves.

The unknown nonlinear system studied in this paper (Figure 1) is a nonlinear Wiener system composed of a linear filter $H(z) = \sum_{k=0}^{N_h-1} h_k z^{-k}$ followed by a zero-memory nonlinearity $g(\cdot)$. This nonlinear system structure has been used in many applications, for example, in satellite communications where the uplink channel is composed of a linear filter followed by a traveling wave tube (TWT) amplifier [5, 19, 20], in microwave amplifier design when modeling solid-state power amplifiers (SSPAs) [13], in adaptive control of nonlinear systems [9], and in biomedical applications when modeling the relationships between cardiovascular signals [1, 2].

The nonlinear system output signal is corrupted by a zero-mean additive white Gaussian noise $N_0(n)$. It can be expressed at time n as

$$d(n) = g\left(\sum_{k=0}^{N_h-1} h_k x(n-k)\right) + N_0(n). \tag{1}$$

The NN model (Figure 1) is composed of an adaptive filter $Q(z) = \sum_{k=0}^{N_q-1} q_k z^{-k}$ followed by a two-layer (zero-memory) adaptive NN. The two-layer NN is composed of a scalar (real-valued) input, M neurons in the input layer, and a scalar output.

This structure aims at adaptively identifying the linear filter H by the adaptive filter Q, and modeling the nonlinearity $g(\cdot)$ by the zero-memory NN.

The unknown nonlinear system is assumed to be a black box and the learning process is performed using the input-output signals only (i.e., the filter-memoryless nonlinearity structure is known, but we do not have access to the internal signals of this structure).

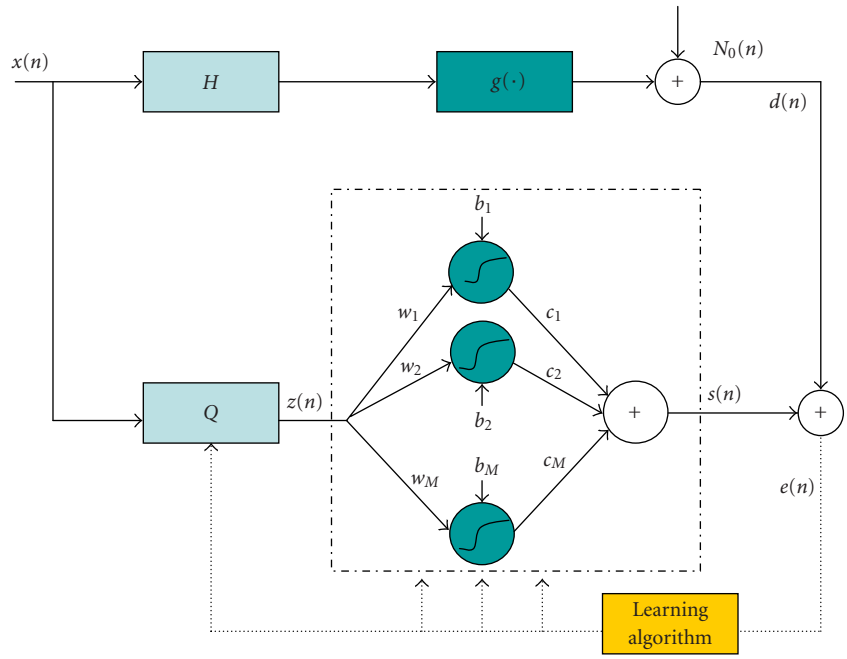


Figure 1: Identification of a nonlinear system with memory using an adaptive neural network.

The NN output at time n is expressed as

$$s(n) = \sum_{k=1}^{M} c_k f\left(w_k \sum_{i=0}^{N_q-1} q_i x(n-i) + b_k\right) = \sum_{k=1}^{M} c_k f\left(w_k Q^t X(n) + b_k\right), \tag{2}$$

where $\{w_k\}$, $\{b_k\}$, $\{c_k\}$, $k = 1,\ldots,M$, are the NN weights, $Q = [q_0 \;\; q_1 \;\; \cdots \;\; q_{N_Q-1}]^t$, and

$$X(n) = [x(n) \;\; x(n-1) \;\; \cdots \;\; x(n-N_Q+1)]^t. \tag{3}$$

The network and filter parameters are updated in order to minimize the loss function l (or squared error) between the system output and the NN output:

$$l(\theta(n)) = \frac{1}{2}e(n)^2 = \frac{1}{2}(d(n)-s(n))^2, \tag{4}$$

where θ represents the set of the adaptive parameters:

$$\theta = \left[w_1 w_2 \cdots w_M \;\; b_1 b_2 \cdots b_M \;\; c_1 c_2 \cdots c_M \;\; q_0 q_1 \cdots q_{N_Q-1}\right]^t. \tag{5}$$

Different NN algorithms are presented and tested in this paper. These algorithms are based on the NG descent, the ordinary gradient descent, and on the Levenberg-Marquardt (LM) procedure [4]. In this paper, we show that the NG learning overcomes the other methods in terms of convergence speed and MSE performance.

We also study which part of the adaptive system (i.e., the linear or nonlinear part) is the most sensitive to the NG descent (in terms of performance improvement).

Comparisons between classical NN and other adaptive system identification approaches are not within the scope of this paper. These comparisons have been extensively studied by several authors (see, e.g., [12] for an extended bibliography). Other applications of the NG descent to satellite communications, such as nonlinear channel predistortion, equalization, and maximum likelihood receiver design can be found in [7].

The following section presents the different NN algorithms. Section 3 presents computer simulations and illustrations. Finally, the conclusion is given in Section 4.

2. ALGORITHMS

2.1. The LMS-backpropagation (LMS-BP) algorithm

The LMS-BP algorithm updates the weights by following the ordinary gradient descent of the error surface:

$$\theta(n+1) = \theta(n) - \mu\nabla_\theta l(\theta(n)), \tag{6}$$

where μ is a small positive constant and ∇_θ represents the ordinary gradient with respect to vector θ, which is expressed as

$$\nabla_\theta l(\theta(n)) = -e(n)\nabla_\theta s(n), \tag{7}$$

where

$$\nabla_\theta s(n) = \begin{pmatrix} c_1 Q^t X(n) f'(w_1 Q^t X(n) + b_1) \\ \vdots \\ c_M Q^t X(n) f'(w_M Q^t X(n) + b_M) \\ c_1 f'(w_1 Q^t X(n) + b_1) \\ \vdots \\ c_M f'(w_M Q^t X(n) + b_M) \\ f(w_1 Q^t X(n) + b_1) \\ \vdots \\ f(w_M Q^t X(n) + b_M) \\ x(n) \sum_{k=1}^{M} c_k w_k f'(w_k Q^t X(n) + b_k) \\ \vdots \\ x(n - N_Q + 1) \sum_{k=1}^{M} c_k w_k f'(w_k Q^t X(n) + b_k) \end{pmatrix}. \tag{8}$$

(In the right-hand side of (8), the weights are at time n.)

This algorithm will be called LMS-BP because the update of the NN weights $\{w_1 w_2 \cdots w_M \;\; b_1 b_2 \cdots b_N \;\; c_1 c_2 \cdots c_M\}$ in (6) corresponds to the classical BP algorithm [14], and the update of the filter weights $\{q_1 q_2 \cdots q_{N_Q}\}$ corresponds to the LMS algorithm [15].

2.2. Natural gradient (NG) learning

The ordinary gradient is the steepest descent direction of a cost function if the space of parameters is an orthonormal coordinate system. It has been shown [17, 21] that in the case of multilayer NNs, the steepest descent direction (or the NG) of the loss function is actually given by

$$-\tilde{\nabla}_\theta l(\theta) = -\Gamma^{-1} \nabla_\theta l(\theta), \tag{9}$$

where Γ^{-1} is the inverse of the Fisher information matrix (FIM)

$$\Gamma = [\gamma_{i,j}] = \left[E\left(\frac{\partial l(s \setminus x; \theta)}{\partial \theta_i} \frac{\partial l(s \setminus x; \theta)}{\partial \theta_j} \right) \right]. \tag{10}$$

Thus, the NG learning algorithm updates the parameters as

$$\theta(n+1) = \theta(n) - \mu \Gamma^{-1} \nabla_\theta l(\theta(n)). \tag{11}$$

The calculation of the expectations in the FIM requires the knowledge of the pdfs of x and s, which are not always available. Moreover, the calculation of the inverse of the FIM is computationally very costly. A Kalman filter technique will be used for an online estimation of the FIM inverse

$$\begin{aligned} &\hat{\Gamma}^{-1}(n+1) \\ &= (1+\varepsilon_n)\hat{\Gamma}^{-1}(n) - \varepsilon_n \hat{\Gamma}^{-1}(n) \nabla_\theta s(n) (\nabla_\theta s(n))^t \hat{\Gamma}^{-1}(n), \end{aligned} \tag{12}$$

where $\nabla_\theta s(n)$ is the (ordinary) gradient of $s(n)$ (see(8)).

A search-and-converge schedule will be used for ε_n in order to obtain a good trade-off between convergence speed and stability:

$$\varepsilon_n = \frac{\varepsilon_0 + c_\varepsilon n/\tau}{1 + c_\varepsilon n/\tau \varepsilon_0 + n^2/\tau} \tag{13}$$

such that small n corresponds to a "search" phase (ε_n is close to ε_0) and large n corresponds to a "converge" phase (ε_n is equivalent to c_ε/n for large n), where ε_0, c_ε, and τ are positive real constants. Using this online Kalman filter technique, the update of the weights (i.e., (11)) becomes

$$\theta(n+1) = \theta(n) - \mu \hat{\Gamma}^{-1} \nabla_\theta l(\theta(n)). \tag{14}$$

This algorithm will be called the coupled NGLMS-NGBP because the filter parameter space and the NN parameter space together are considered as a single space.

2.3. The disconnected NGLMS-NGBP algorithm

Since the filter and the memoryless NN are physically separated, then a choice can be made concerning the parameter space:

(i) either we consider a single parameter space for the filter coefficients and NN weights (as we have done above),

(ii) or we consider two different parameter spaces, one for the filter and the other for the neural network. In this case, the parameter space of filter Q can be described with an FIM $\Gamma_2 = [\gamma_{i,j}(Q)]$ which equals

$$\gamma_{i,j}(Q) = E\left[\frac{\partial l(s|x;Q)}{\partial q_i} \frac{\partial l(s|x;Q)}{\partial q_j} \right]. \tag{15}$$

The parameter space for the NN is described by a new vector

$$\theta_{NN} = \left[w_1 w_2 \cdots w_M \;\; b_1 b_2 \cdots b_M \;\; c_1 c_2 \cdots c_M \right]^t, \tag{16}$$

and its FIM will be denoted as Γ_1.

The same Kalman filter technique as in Section 2.2 will be used here in order to avoid the explicit calculation of the inverses of the two FIMs, Γ_1^{-1} and Γ_2^{-1}, which will be estimated online by $\hat{\Gamma}_1^{-1}$ and $\hat{\Gamma}_2^{-1}$ as follows:

$$\begin{aligned} \hat{\Gamma}_1^{-1}(n+1) &= (1+\varepsilon_n)\hat{\Gamma}_1^{-1}(n) \\ &\quad - \varepsilon_n \hat{\Gamma}_1^{-1}(n) (\nabla_{\theta_{NN}} s) (\nabla_{\theta_{NN}} s)^T \hat{\Gamma}_1^{-1}(n), \\ \hat{\Gamma}_2^{-1}(n+1) &= (1+\varepsilon_n)\hat{\Gamma}_2^{-1}(n) \\ &\quad - \varepsilon_n \hat{\Gamma}_2^{-1}(n) (\nabla_Q s) (\nabla_Q s)^T \hat{\Gamma}_2^{-1}(n). \end{aligned} \tag{17}$$

The adaptive system parameters are therefore updated as follows:

$$\begin{pmatrix} \theta_{NN}(n+1) \\ Q(n+1) \end{pmatrix} = \begin{pmatrix} \theta_{NN}(n) \\ Q(n) \end{pmatrix} - \mu \begin{pmatrix} \hat{\Gamma}_1^{-1}(n) \nabla_{\theta_{NN}} l(\theta_{NN}) \\ \hat{\Gamma}_2^{-1}(n) \nabla_Q l(Q) \end{pmatrix}, \tag{18}$$

where $\hat{\Gamma}_1^{-1}$ is the inverse Fisher matrix for the NN weights (a $3M \times 3M$ matrix) and $\hat{\Gamma}_2^{-1}$ is the inverse Fisher matrix for the filter weights (an $N_Q \times N_Q$ matrix). The other terms are expressed as

$$\nabla_{\theta_{NN}} l(\theta_{NN}) = -e(n)\nabla_{\theta_{NN}} s(n), \tag{19}$$

where

$$\nabla_{\theta_{NN}} s(n) = \begin{pmatrix} c_1 Q^t X(n) f'(w_1 Q^t X(n) + b_1) \\ \vdots \\ c_M Q^t X(n) f'(w_M Q^t X(n) + b_M) \\ c_1 f'(w_1 Q^t X(n) + b_1) \\ \vdots \\ c_M f'(w_M Q^t X(n) + b_M) \\ f(w_1 Q^t X(n) + b_1) \\ \vdots \\ f(w_M Q^t X(n) + b_M) \end{pmatrix} \tag{20}$$

and $\nabla_Q l(Q) = -e(n)\nabla_Q s(n)$, where

$$\nabla_Q s(n) = \left(\sum_{k=1}^{M} c_k w_k f'(w_k Q^t X(n) + b_k) \right) X(n). \tag{21}$$

This algorithm will be called the separated (or disconnected) NGLMS-NGBP algorithm because the two spaces are treated separately. Note that the computational complexity is lower than the single space NG algorithm because here we deal with two small matrices rather than a large one (i.e., this is equivalent to neglecting the coupling terms between the filter and the NN in the matrix Γ). In the simulations below, we will show that these terms are negligible in practice (see Figure 2).

Other variants of this algorithm can be derived easily, depending on where we would like to apply the NG procedure. For example, if we would like to use the classical LMS algorithm for the adaptive filter and use the NG for the NN, then we keep the upper equations in (18) which concern the update of θ_{NN} and use the LMS algorithm for Q (i.e., the update of filter Q in (6)).

3. ILLUSTRATIONS AND SIMULATION RESULTS

3.1. Description of the unknown system and the algorithms that have been implemented

Concerning the unknown structure to be identified, we have taken the nonlinearity as

$$g(x) = \frac{\alpha x}{1 + \beta x^2}, \quad \alpha = 2,\ \beta = 1. \tag{22}$$

This function has the same shape as amplitude conversions of several high-power amplifiers used in communications. The input signal has been taken as a white Gaussian sequence with variance 1.

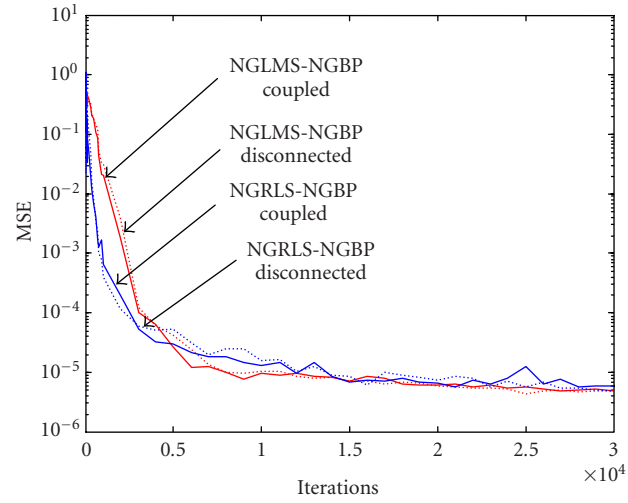


Figure 2: Learning curves of the coupled and disconnected NG algorithms.

The filter H weights were taken as

$$H = [1.4961 \quad 1.3619 \quad 1.0659 \quad 0.6750 \quad 0.2738 \quad -0.0575 \quad -0.2636 \quad -0.3257 \quad -0.2634 \quad -0.1250]^t. \tag{23}$$

The noise standard deviation was $\sigma = 0.002$, $\varepsilon_0 = 0.005$, $c_\varepsilon = 1$ and $\tau = 70,000$ (for the NG algorithms), and $\lambda = 0.99$ (for the RLS algorithm). The learning rate was fixed to $\mu = 0.007$ (for each algorithm), this is because this value represents a good trade-off between convergence speed and MSE error. Fifty Monte Carlo runs have been done to estimate the learning curves.

The filter-nonlinearity structure corresponds to a typical model of uplink satellite channels [5]. The filter Q is composed of 10 weights which have been initialized with 0. The NN was composed of $M = 5$ neurons which have been initialized with small random values (the same values have been taken for the different algorithms, so that the initial point in the MSE surface is the same for all algorithms). The number of neurons has been chosen equal to 5 because a higher number does not significantly improve the approximation of the nonlinearity, whereas a lower number strongly affects the approximation performance.

The purpose of this part is to study the efficiency of natural gradient learning and to see which part of the adaptive system is the most sensitive to NG learning (in terms of improvement of the algorithm performance).

We have implemented the following algorithms:

(i) the classical LMS algorithm for the adaptive filter and the BP for the NN (LMS-BP),

(ii) the classical LMS for the adaptive filter and the NG for the NN (LMS-NGBP),

(iii) NG LMS for the adaptive filter and BP for the NN (NGLMS-BP),

(iv) NG LMS for the adaptive filter and NG for the NN, the parameter space is considered as a single space as explained in Section 2.2 (coupled NGLMS-NGBP),

(v) NG LMS for the adaptive filter and NG for the NN. Both algorithms are separated in the sense of Section 2.3 (NGLMS-NGBP, disconnected).

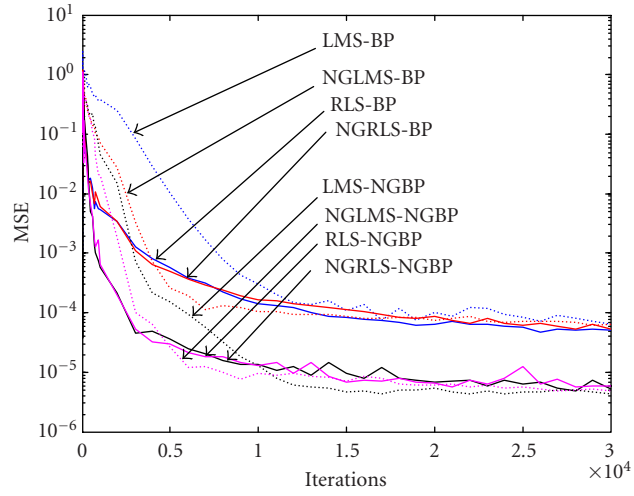
We have also implemented the RLS algorithm [15] for the adaptive filter (instead of LMS) and tested the following algorithms: RLS-BP, RLS-NGBP, NGRLS-BP, NGRLS-NGBP (coupled), and NGRLS-NGBP (disconnected). See Appendix A for the NGRLS-NGBP algorithm.

The performance of our system has also been compared to the LM procedure (see, e.g., [4]), which has been adapted to our identification problem (see Appendix B). Similarly to the NG algorithm, the LM algorithm can be applied to the whole structure (we will call it LMLMS-LMBP), or to the nonlinear part only (we will call it LMS-LMBP). The parameter λ for the LM algorithm was initialized by $\lambda(1) = 0.2$, then, every 50 iterations, it is divided or multiplied by 5, depending on whether the average squared error has decreased or increased during the last 50-iteration window.

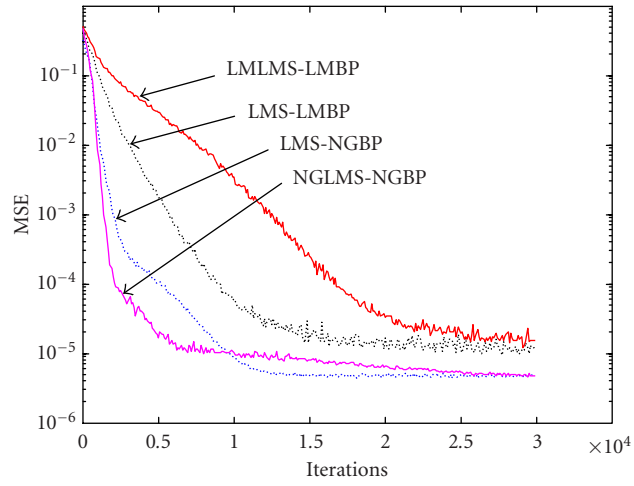
3.2. System identification and sensitivity to natural gradient descent

Figure 2 compares the learning curves obtained by coupled and separated versions of the NGLMS-NGBP algorithm as well as the NGRLS-NGBP algorithm. It can be seen that there is no significant difference between the coupled and separated versions. This shows that the coupling terms in the FIM inverse can be neglected. This can be explained by the fact that both unknown system and model are composed of two physically separated parts (i.e., a linear filter and a memoryless nonlinearity). Therefore, the coupling terms are not expected to significantly affect the convergence speed. This certainly depends on the system model, for example, if the unknown system and/or the adaptive model are not composed of physically separated parts, then the coupling terms may play an important role in the convergence behavior. The authors in [1, 2, 6, 13] have presented interesting discussions on the BP algorithm applied to nonlinear Wiener and Hammerstein systems and have given some analytical and qualitative results on the convergence behavior of each part of the adaptive system. Therefore, in what follows, we will keep the separated version of the algorithms which is computationally less complex.

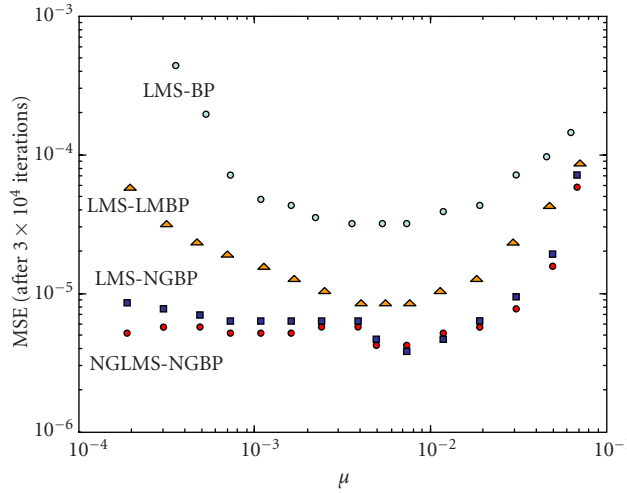
Figure 3a compares the learning curves of 8 versions of the NG algorithm. It can be seen that, in order to obtain an improvement of the convergence speed and MSE performance, the NG descent should be applied at least to the NN part. If the ordinary gradient is used for the NN, then the algorithm may be trapped in a local minimum, whether we apply the NG to the filter or not. When the NG is applied to both parts (i.e., the filter Q and the NN), then there is only a slight improvement compared to applying the NG to the NN part only. This can be explained by the fact that the linear part converges very quickly to a scaled version of the



(a) Learning curves: comparison between natural gradient and ordinary gradient.



(b) Learning curves: comparison between the NG and LM approaches.



(c) MSE performance versus learning rate μ.

FIGURE 3

Table 1 Generalization MSE for the different algorithms.

Algorithm	Generalization MSE
LMS-BP	$5.2\ 10^{-5}$
NGLMS-BP	$5.1\ 10^{-5}$
RLS-BP	$4.9\ 10^{-5}$
NGRLS-BP	$4.8\ 10^{-5}$
LMS-NGBP	$4.8\ 10^{-6}$
RLS-NGBP	$4.9\ 10^{-6}$
NGLMS-NGBP	$5\ 10^{-6}$
NGRLS-NGBP	$4.9\ 10^{-6}$
LMLMS-LMBP	$1.8\ 10^{-5}$
LMS-LMBP	$1.2\ 10^{-5}$

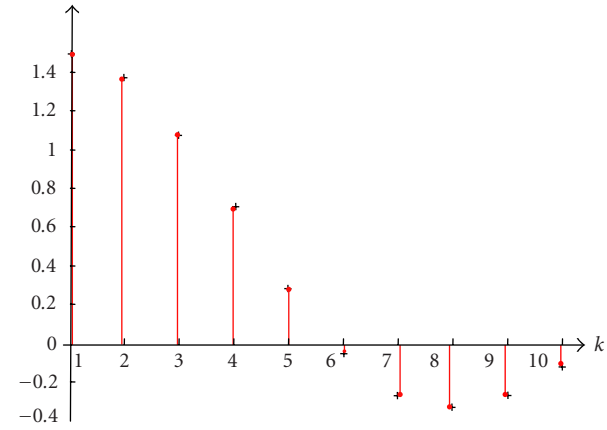


Figure 4: Impulse responses of the adaptive filters after normalization and unknown filter H, where • represents q_k and + represents h_k. (All impulse responses are almost superimposed.)

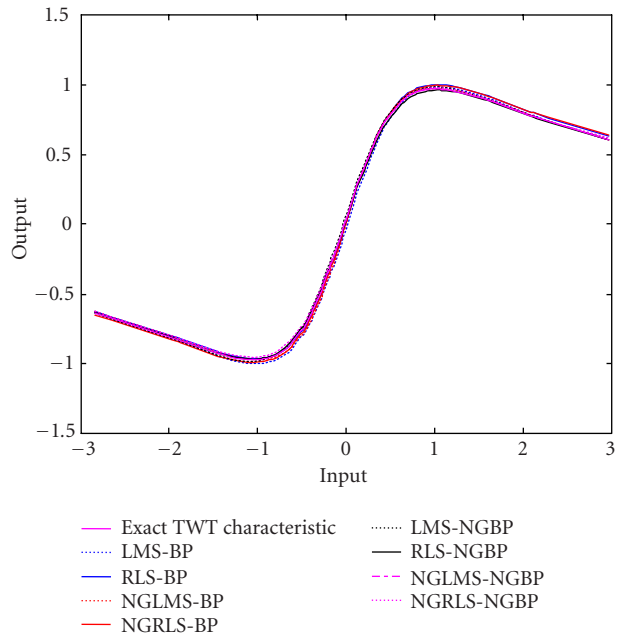


Figure 5: Transfer functions of the converged nonlinear memoryless parts obtained by the different algorithms (after normalization) and comparison with the exact TWT characteristic.

unknown filter, whereas the nonlinear part takes more time to converge. We can conclude that the overall convergence speed is mostly controlled by the nonlinear part and that the NG descent improvement will mostly affect this part of the system.

This means that it is more interesting to apply NG to the NN part only rather than applying it to the whole structure. This considerably reduces the computational complexity while keeping a good overall performance. For example, Figure 3 shows that, in order to achieve an MSE of 10^{-4}, the LMS-NGBP needs 6,000 iterations, whereas the LMS-BP needs more than 17,000 iterations.

Figure 3b compares the NG to the LM algorithm. The LMS-LMBP and the LMLMS-LMBP have been tested. It can be seen that the NG algorithm outperforms these two versions of the LM algorithm. For example, an MSE error of 10^{-4} is reached by the NGLMS-NGBP algorithm in less than 5,000 iterations, whereas the same error is reached by the LMS-LMBP in 10,000 iterations, and by the LMLMS-LMBP in 17,000 iterations. The LM final MSE is also higher than that of the NG algorithm (see Table 1).

Concerning the effect of the learning rate μ on the convergence speed, Figure 3c illustrates the MSE error (at the 30,000th iteration) versus μ for three different algorithms (the other algorithms were not shown for the clarity of the figure). It can be seen that the value 0.007 represents an optimal value for the MSE performance. The LMS-BP has a typical behavior of an ordinary gradient-type descent, with a global minimum and an increasing MSE as we go away from the minimum. This is because large μ introduces more error, and for small μ, the convergence is slow, so the MSE error reached at the 30,000th iteration remains high. For the LMS-NGBP, the optimal value is 0.007, however, the MSE curve is very flat for smaller values (which means that the algorithm converged well before the 30,000th iteration). For high μ, the MSE increases relatively fast, and instability may occur. Finally, for the LMS-LMBP, the MSE curve behavior is somewhat between the LMS-BP and LMS-NGBP, which is expected since, as discussed in Appendix B, the LM algorithm has a mechanism which is a kind of "combination" between the NG descent and the ordinary descent.

Computer simulations show that Q converges to H (within a scale factor) for all algorithms. In Figure 4 we have superimposed the impulse response of filter H and that of the converged filter Q for each of the above algorithms (after normalization with the inverse of the scale factors).

Concerning the nonlinearity, it has also been successfully identified by the memoryless NN. Figure 5 superimposes the unknown nonlinearity $g(z)$ and the NN transfer functions obtained by the different algorithms. It can be seen that all these functions are close to $g(z)$.

Table 1 gives the generalization MSE (i.e., MSE obtained by the different converged structures for an input that was not used in the learning process). It can be seen that the NG approach yields better MSE approximation performance than the other methods.

3.3. Tracking capabilities

In order to illustrate the tracking capabilities of the algorithms, we simulated a change in the nonlinearity $g(\cdot)$ occurring during online learning at the 25,000th iteration. Figure 6a shows the old and new characteristic of the nonlinearity. This may happen for example in satellite communications where TWT amplifier characteristics are subject to change because of thermodynamical perturbations.

It can be seen from Figure 6b that, when the change occurs, the MSE considerably increases, then it is decreased by the algorithm. The NG approach again is much faster to track the change than the ordinary gradient. The final MSE is also better (the ordinary gradient algorithms stood in local minima until the end of the learning process). (The LM procedure is not included in these comparisons since it has a lower performance than the NG algorithm for a higher computational complexity.)

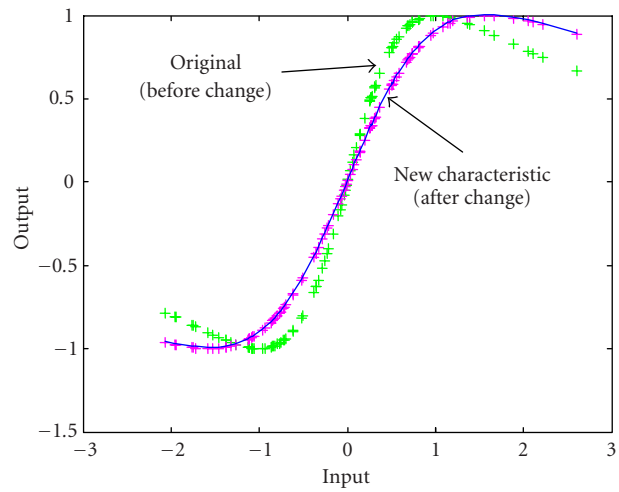
Figure 7 shows the sensitivity of applying the NG to the linear part (see also [18] for an interesting study of the NG algorithm applied to linear systems). The figure shows that, even though the LMS-NGBP algorithm gives a smaller MSE (before the change) than the NGLMS-NGBP algorithm, it is slower to track the change. This can be explained by the fact that the change that affected the nonlinearity has also introduced a considerable misadjustment error on the filter weights pushing them far away from the point that was reached before the change. The weights had then to be reupdated in order to reach their original state before the change (see also [6] for a detailed analytical study in the case of the BP algorithm). Other simulation results show that the algorithms are robust to changes (in both linear and nonlinear parts), that is, even for more severe changes, the tracking capabilities are good and the NG descent always outperforms the ordinary gradient descent.

It should be noted that the NG variants of the RLS-BP algorithm have, in general, a slightly better convergence speed than the NG variants of the LMS-BP algorithm, except for the tracking problem (Figure 6b), where the NGLMS-NGBP algorithm was faster to track the change than the NGRLS-NGBP algorithm.

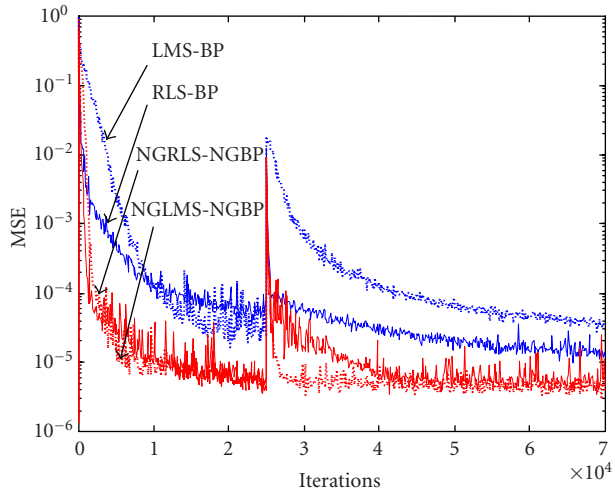
3.4. Computational complexity

Although the NG learning outperforms the BP algorithm in terms of convergence speed and MSE, it has a higher computational complexity. Here we compare only the BP and NG applied to the nonlinear part (composed of M neurons) since we have seen that it is not worth using the NG for the linear part. For a network composed of M neurons, the BP algorithm requires $O(M)$ multiplications and additions at each iteration. When the NG is used, this value becomes $O(M^2)$. This is due to the matrix multiplication occurring in the NG weight update and in the estimation of the inverse of the Fisher matrix.

One of the advantages of the BP algorithm is that we can exploit parallelism between neurons. Thus, with M processors working in parallel, we can reach a complexity of $O(1)$ per processor for each iteration. For the NG algorithm, parallelism can also be exploited: with M parallel processors, each



(a) Nonlinearity before and after the change.



(b) Tracking of change in the nonlinear part: learning curves of the natural gradient and ordinary gradient.

Figure 6

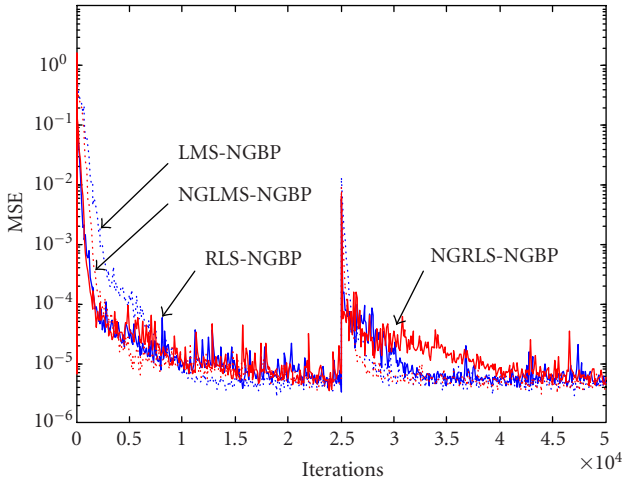


FIGURE 7: Tracking of change in the nonlinear part: sensitivity of the linear part to the NG descent.

iteration will require $O(M)$ multiplications and additions per processor. The number of times we need to calculate the output of the sigmoid is, of course, the same for both algorithms.

4. CONCLUSION

This paper studied different NG descent-based NN algorithms for the identification of nonlinear systems with memory. The unknown system was comprised of a linear filter followed by a memoryless nonlinearity. The NN structure was composed of a linear adaptive filter followed by a memoryless nonlinear NN. A Kalman filter-based technique and a search-and-converge method have been employed for the NG algorithm. Computer simulations have shown that the NG approach gives faster convergence speed and smaller MSE than the ordinary gradient descent and the LM procedure. Several variants of the NG algorithm have been studied. It has been shown that the improvement of the convergence speed is mainly governed by applying the NG to the nonlinear NN part (i.e., applying the NG to both linear and nonlinear parts does not significantly improve the performance, compared to applying the NG to the nonlinear NN part only). An application of tracking of changes in nonlinear systems has also been presented.

APPENDICES

A. NGRLS-NGBP ALGORITHM

We use here the estimation of the steepest descent to calculate the gain and the estimated covariance matrix at each step in the recursive least square (RLS [15]) algorithm. We have again two estimations of the inverse of Fisher matrices, one for the perceptron parameters and one for the filter coefficients. We simply replace the classic gradient by the steepest descent estimation in the classic RLS equations. This gives

$$\begin{aligned}
\begin{pmatrix}\theta_{NN}(n+1)\\ Q(n+1)\end{pmatrix} &= \begin{pmatrix}\theta_{NN}(n)\\ Q(n)\end{pmatrix} - e(n)\begin{pmatrix}\mu\hat{\Gamma}_1^{-1}(n)\nabla_{\theta_{NN}}s,\\ K(n)\end{pmatrix},\\
K(n+1) &= \frac{\hat{P}(n)\hat{\Gamma}_2^{-1}(n)\nabla_Q s}{\lambda + (\nabla_Q s)^T\hat{\Gamma}_2^{-1}(n)\hat{P}_n\hat{\Gamma}_2^{-1}(n)\nabla_Q s},\\
\hat{P}(n+1) &= \frac{1}{\lambda}\Big(\hat{P}(n) - K(n)(\nabla_Q s)^T\hat{\Gamma}_2^{-1}(n)\hat{P}(n)\Big),\\
\hat{\Gamma}_1^{-1}(n+1) &= (1+\varepsilon_n)\hat{\Gamma}_1^{-1}(n)\\
&\quad - \varepsilon_n\hat{\Gamma}_1^{-1}(n)(\nabla_{\theta_{NN}}s)(\nabla_{\theta_{NN}}s)^T\hat{\Gamma}_1^{-1}(n),\\
\hat{\Gamma}_2^{-1}(n+1) &= (1+\varepsilon_n)\hat{\Gamma}_2^{-1}(n)\\
&\quad - \varepsilon_n\hat{\Gamma}_2^{-1}(n)(\nabla_\theta s)(\nabla_\theta s)^T\hat{\Gamma}_2^{-1}(n).
\end{aligned} \tag{A.1}$$

Note that we can consider also the whole space

$$\theta = \Big[w_1 w_2 \cdots w_M \;\; b_1 b_2 \cdots b_M \;\; c_1 c_2 \cdots c_M \;\; q_0 q_1 \cdots q_{N_Q-1}\Big]^t, \tag{A.2}$$

and derive the single space NGRLS-NGBP algorithm as we did in Section 2.1.

B. THE LEVENBERG-MARQUARDT (LM) ALGORITHM

The LM learning algorithm (see, e.g., [4]) can be extended to our system identification problem by updating the parameters as follows:

$$\theta(n+1) = \theta(n) - \mu G^{-1}\nabla_\theta l(\theta(n)), \tag{B.1}$$

where $G^{-1} = [\nabla_\theta l(\theta(n)) \cdot \nabla_\theta l(\theta(n))^t + \lambda(n)I]^{-1}$, with I the identity matrix and $\lambda(n)$ a positive scalar which is decreased after each reduction of the cost function (e.g., by dividing λ by 5) and is increased only when a tentative step would increase the cost function (e.g., multiplying λ by 5).

One of the main difficulties in the implementation of the LM method is an effective strategy for controlling the size of λ at each iteration. Some researchers propose a method, which is to estimate the relative nonlinearity using a linear prediction and a cubic interpolated estimation. In our implementation, we simplify the method by evaluating the cost function over a window of p iterations, for example, $p = 50$, and λ is updated every p iterations.

Another main difficulty is the problem of an ill-conditioned matrix when we make the inverse operation. We will then use the Kalman filtering technique to estimate it online:

$$\begin{aligned}
&\hat{G}^{-1}(n+1)\\
&\quad = (1+\varepsilon(n))\hat{G}^{-1}(n)\\
&\qquad - \varepsilon(n)\hat{G}^{-1}(n)[\nabla_\theta s(n)(\nabla_\theta s(n))' + \lambda(n)I]\hat{G}^{-1}(n).
\end{aligned} \tag{B.2}$$

The updating procedure then becomes

$$\theta(n+1) = \theta(n) - \mu\hat{G}^{-1}\nabla_\theta l(\theta(n)). \tag{B.3}$$

The LM approach can be seen as a combination between the NG and the ordinary gradient descent. When λ is small, it follows the NG descent. When λ is high, it follows the ordinary gradient descent.

Similarly to the NG algorithm, the LM approach can be applied to both linear and nonlinear parts. Also the space of parameters can be considered as a whole space or as two separated spaces (one for the linear part, the other for the nonlinear part). In our simulations, we have taken the same Kalman procedure parameters for both NG and LM algorithms.

Note that the LM approach is more complex than the NG algorithm because of the conditional update of λ and the matrix addition in (B.2).

ACKNOWLEDGMENTS

This work was supported by the Canadian Institute for Telecommunications Research/the Canadian Space Agency (CITR/CSA), the Natural Sciences and Engineering Research Council (NSERC), and the Premier's Research Excellence Award (PREA), Ontario, Canada.

REFERENCES

[1] N. J. Bershad, P. Celka, and J. M. Vesin, "Stochastic analysis of gradient adaptive identification of nonlinear systems with memory for Gaussian data and noisy input and output measurements," *IEEE Trans. Signal Processing*, vol. 47, no. 3, pp. 675–689, 1999.

[2] N. J. Bershad, P. Celka, and J. M. Vesin, "Analysis of stochastic gradient tracking of time-varying polynomial Wiener systems," *IEEE Trans. Signal Processing*, vol. 18, no. 6, pp. 1676–1686, 2000.

[3] J. Sjoberg, Q. Zhang, and L. Ljung, "Nonlinear black box modeling in system identification: a unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.

[4] L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, Upper Saddle River, NJ, USA, 1999.

[5] S. Benedetto and E. Biglieri, *Principles of Digital Transmission with Wireless Applications*, Kluwer Academic Press, Norwell, Mass, USA, 1999.

[6] M. Ibnkahla, "Statistical analysis of neural network modeling and identification of nonlinear channels with memory," *IEEE Trans. Signal Processing*, vol. 50, no. 6, pp. 1508–1517, 2002.

[7] M. Ibnkahla, J. Yuan, and R. Boutros, "Neural networks for transmissions over satellite channels," to appear in *Signal Processing for Mobile Communications Handbook*, M. Ibnkahla, Ed., CRC Press, Boca Raton, Fla, USA, 2004.

[8] M. Ghogho, S. Meddeb, and J. Bakkoury, "Identification of time-varying nonlinear channels using polynomial filters," in *Proc. IEEE Workshop on Nonlinear Signal and Image Processing (NSIP '97)*, Mackinac Island, Mich, USA, September 1997.

[9] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[10] S. Prakriya and D. Hatzinakos, "Blind identification of LTI-ZMNL-LTI nonlinear channel models," *IEEE Trans. Signal Processing*, vol. 43, no. 12, pp. 3007–3013, 1995.

[11] J. Ralston, A. Zoubir, and B. Bouashash, "Identification of a class of nonlinear systems under stationary non-Gaussian excitation," *IEEE Trans. Signal Processing*, vol. 45, no. 3, pp. 719–735, 1997.

[12] M. Ibnkahla, "Applications of neural networks to digital communications: A survey," *Signal Processing*, vol. 80, no. 7, pp. 1185–1215, 2000.

[13] M. Ibnkahla, N. J. Bershad, J. Sombrin, and F. Castanié, "Neural network modeling and identification of nonlinear channels with memory: algorithms, applications and analytic models," *IEEE Trans. Signal Processing*, vol. 46, no. 5, pp. 1208–1220, 1998.

[14] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, D. Rumelhart and J. McClelland, Eds., pp. 318–362, M. I. T. Press, Cambridge, Mass, USA, 1986.

[15] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Upper Saddle River, NJ, USA, 1996.

[16] S. Haykin, *Neural Networks: A Comprehensive Foundation*, IEEE Press, New York, NY, USA, 1994.

[17] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.

[18] L.-Q. Zhang, S.-I Amari, and A. Cichocki, "Semiparametric model and superefficiency in blind deconvolution," *Signal Processing*, vol. 81, no. 12, pp. 2535–2553, 2001.

[19] P. Hetrakul and D. Taylor, "The effects of transponder nonlinearity on binary CPSK signal transmission," *IEEE Trans. Communications*, vol. 29, pp. 546–553, May 1976.

[20] A. Saleh, "Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers," *IEEE Trans. Communications*, vol. 29, no. 11, pp. 1715–1720, 1981.

[21] S.-I. Amari, H. Park, and K. Fukumizu, "Adaptive method of realizing natural gradient learning for multilayer perceptrons," *Neural Computation*, vol. 12, no. 6, pp. 1399–1409, 2000.

Mohamed Ibnkahla received the Ph.D. degree in 1996 and the HDR (the ability to lead research) degree in 1998 from the National Polytechnic Institute of Toulouse (INPT), France. He is currently an Assistant Professor in the Department of Electrical and Computer Engineering at Queen's University, Kingston, Canada. He is the editor of *Signal Processing for Mobile Communications Handbook*, CRC Press (to appear in 2004). Dr. Ibnkahla received the INPT Leopold Escande Medal for the year 1997, for his research contributions to signal processing, and the Prime Minister's Research Excellence Award (PREA), Province of Ontario, Canada, in 2000, for his contributions in wireless mobile communications.