COLLABORATIVE ADAPTIVE LEARNING USING HYBRID FILTERS

D. Mandic[1]*, P. Vayanos*[1]*, C. Boukis*[2]*, B. Jelfs*[1]*, S.L. Goh*[1]*, T. Gautama*[3] *and T. Rutkowski*[4]

[1]Imperial College London, UK, [2]AIT Athens, Greece, [3]Phillips Leuven, Belgium, [4]BSI RIKEN, Japan

E-mail: {b.jelfs,foivi.vayanos,su.goh,d.mandic}@imperial.ac.uk, cbou@ait.edu.gr, temujin.gautama@philips.com, tomek@brain.riken.jp

ABSTRACT

A novel stable and robust algorithm for training of finite impulse response adaptive filters is proposed. This is achieved based on a convex combination of the Least Mean Square (LMS) and a recently proposed Generalised Normalised Gradient Descent (GNGD) algorithm. In this way, the desirable fast convergence and stability of GNGD is combined with the robustness and small steady state misadjustment of LMS. Simulations on linear and nonlinear signals in the prediction setting support the analysis.

Index Terms— Distributed, adaptive, collaborative SP

1. INTRODUCTION

The Least Mean Square (LMS) algorithm is a de facto standard for training of adaptive finite impulse response (FIR) filters, and can be described by [1]

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

where $e(k)$ is the instantaneous error at the output of the filter for the time instant k, $d(k)$ is the desired signal, $\mathbf{x}(k) = [x(k-1), \ldots, x(k-N)]^T$ is the input signal vector, N is the length of the filter, $(\cdot)^T$ denotes the vector transpose operator, and $\mathbf{w}(k) = [w_1(k), \ldots, w_N(k)]^T$ is the filter coefficient (weight) vector. The parameter μ (the step–size) is critical to the convergence and dynamical behaviour of LMS.

Despite the small steady–state misadjustment and robustness exhibited by the LMS, its low convergence speed has initiated research on faster algorithms within the same class. Ideally, we desire an algorithm which exhibits fast convergence and small steady state misadjustment when operating in a statistically stationary environment, whereas when operating in a statistically nonstationary environment the algorithm should adapt according to the dynamics of the input signal.

One such algorithm is the normalised LMS (NLMS) [1, 2] which is faster and more responsive than LMS, but is also prone to poor steady-state performance and divergence for certain classes of inputs. The NLMS weight update can be expressed as

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) + \frac{\mu}{\parallel \mathbf{x}(k) \parallel_2^2} e(k)\mathbf{x}(k) \\ &= \mathbf{w}(k) + \eta(k)e(k)\mathbf{x}(k) \end{aligned} \tag{2}$$

where $\parallel \cdot \parallel_2$ denotes the Euclidean norm. By such normalisation, the error surface defined by the cost function $E(k) = \frac{1}{2}e^2(k)$ is "regularised", and the step size $\eta(k)$ is varied according to the changing power levels of the input signal.

Modifications of LMS which cater for the critical cases of i) inputs with large dynamical ranges, ii) ill-conditioned input autocorrelation matrices and iii) coupling between different signal modes, include classes of algorithms based on a gradient–adaptive step size μ [3], mixed norm algorithms [4], the ε–NLMS class [5] algorithms, and combinations of adaptive filters [6].

Gradient adaptive step size algorithms [7, 3, 8] are based upon estimators of $\partial E(k)/\partial \mu$, which increases their sensitivity not only to the correlation between input signal samples, but also the value of the parameter that governs the adaptation of $\eta(k)$. The idea behind mixed–norm algorithms [4] is to combine the minimum mean square error (MMSE) training defined by $E(k) = \frac{1}{2}e^2(k)$, with other cost functions, such as $E(k) = |e(k)|$, in order to balance between the different error measures. Convex combinations of filters typically employ two LMS–trained filters with different step–sizes, recent results show that such a "hybrid" filter achieves faster convergence and more controlled performance than a single filter [6].

The recently introduced Generalised Normalised Gradient Descent (GNGD) algorithm [5] belongs to the class of ε–NLMS algorithms, for which the NLMS step–size is modified to

$$\frac{\mu}{\parallel \mathbf{x} \parallel_2^2} \rightarrow \frac{\mu}{\parallel \mathbf{x}(k) \parallel_2^2 + \varepsilon} \tag{3}$$

where ε is a small positive "regularisation" constant. The GNGD makes the regularisation term ε within the denominator of the learning rate of NLMS gradient adaptive. This introduces excellent stability, robustess to parameter perturbations, and very fast convergence. However, it was realised that for certain inputs the update of ε from (3) may not settle when in the steady state.

To that cause, we propose to equip the GNGD algorithm with more desirable steady state characteristics. This is achieved by employing a convex combination of filters trained by GNGD and LMS, as shown in Figure 1. Simulation results show that the proposed approach attains the excellent stability and fast convergence of GNGD, together with the desired steady state characteristics of LMS.

1-4244-0728-1/07/$20.00 ©2007 IEEE

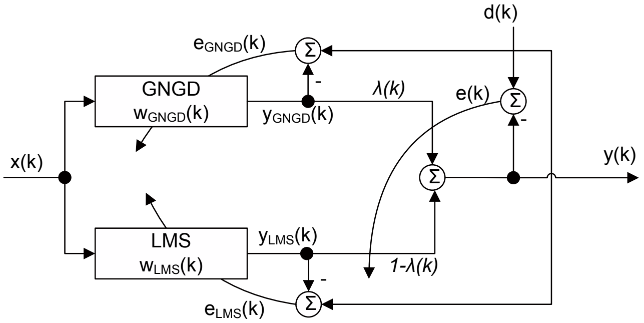

Fig. 1. Convex combination of adaptive filters.

2. GENERALISED NORMALISED GRADIENT DESCENT (GNGD)

The performance of NLMS is governed by the value of its step size μ, for which the stability bounds are $0 < \mu < 2$. In practical applications, small values of $\|\mathbf{x}(k)\|_2^2$ cause effectively an increase in the value of $\eta(k) = \frac{\mu}{\|\mathbf{x}(k)\|_2^2}$, which may bring to poor performance and ultimately divergence of NLMS, when the resulting $\eta(k)$ grows outside the stability bounds, as illustrated in Figure 2.

The GNGD [5] solves this problem by performing a gradient update of the regularisation parameter ε from (3), in the generic form of

$$\varepsilon(k+1) = \varepsilon(k) - \rho\nabla_{\varepsilon(k)}E(k) \qquad (4)$$

where ρ is some small constant. The following expressions describe the operation of GNGD

$$\begin{aligned} y(k) &= \mathbf{x}^T(k)\mathbf{w}(k) \qquad e(k) = d(k) - y(k) \\ &\eta(k) = \frac{\mu}{\| \mathbf{x}(k) \|_2^2 + \varepsilon(k)} \\ \mathbf{w}(k+1) &= \mathbf{w}(k) + \eta(k)e(k)\mathbf{x}(k) \\ \varepsilon(k+1) &= \varepsilon(k) - \rho\mu\frac{e(k)e(k-1)\mathbf{x}^T(k)\mathbf{x}(k-1)}{\left(\| \mathbf{x}(k-1) \|_2^2 + \varepsilon(k-1)\right)^2} \quad (5) \end{aligned}$$

The GNGD has been shown to converge extremely fast, even in environments where NLMS diverges. It is also robust to perturbations of the regularisation term ε and the initialisation of the step size adaptation parameter ρ. The GNGD has the desirable property that it is almost guaranteed *not to diverge*, no matter what the statistical properties of the environment are. This is illustrated in Figure 2 where, in order to simulate the close to zero input to the filter, we employed $\mu = 2.1$, for which NLMS diverged and GNGD converged.

Despite its excellent stability and very fast convergence, GNGD does not guarantee that in the steady state, for some very small output error of the filter, the update of the ε term in (5) will settle to some fixed value. This is due to the fact, that the filter remains "alert" at all time instants, in order to react quickly to the changes in the environment, this can be seen from the update of $\varepsilon(k)$ in (5). This can, in turn, cause an unwanted effect of an increase in the steady state error when the algorithm should effectively be at the steady-state.

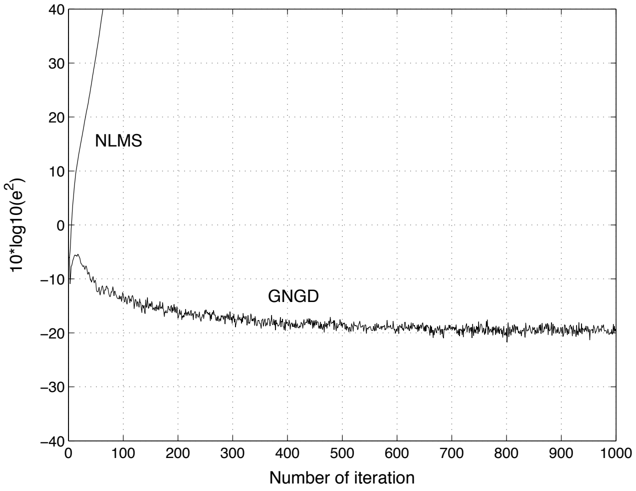

Fig. 2. Performance of NLMS and GNGD for $\mu = 2.1$.

One way to improve the steady state performence of GNGD is to employ a "search then converge" (STC) scheme [9]. There, the learning rate $\eta(k)$ is modified according to

$$\eta(k) = \frac{\mu}{\| \mathbf{x}(k) \|_2^2 + \varepsilon(k) + STC(k)} \qquad (6)$$

where the term $STC(k)$ refers to a cooling schedule [10]. This however proves impractical when we do not possess prior knowledge about the statistics of the input signals, and limits the application of this approach.

3. THE PROPOSED APPROACH

We desire to make the GNGD algorithm achieve better steady state performance, while keeping its fast initial convergence and excellent tracking capabilities. Following the approach from [6], we propose to achieve this by a combination of two adaptive filters trained respectively by GNGD and LMS, whose outputs are then combined in a convex manner to form $y(k)$, as shown in Figure 1.

From Figure 1, the overall output $y(k)$ of such a scheme can be expressed as

$$y(k) = \lambda(k)y_{GNGD}(k) + (1 - \lambda(k))y_{LMS}(k) \qquad (7)$$

where $y_{GNGD}(k)$ and $y_{LMS}(k)$ are respectively the outputs of the GNGD– and LMS-trained subfilters. Our aim is to make the value of mixing parameter λ within this structure *adapt* according to the dynamics of the input signal, so as to benefit from the fast convergence and stability of GNGD and steady state performance of LMS.

The idea is that in the beginning of the adaptation, and for sudden statistical changes in the enviroment, the values of λ are such that the overall output $y(k)$ is dominated by $y_{GNGD}(k)$, whereas in the steady state, it should become closer to $y_{LMS}(k)$.

The weight vectors $\mathbf{w}_{GNGD}$ and $\mathbf{w}_{LMS}$ are updated independently based on their corresponding output errors $e_{GNGD}(k)$ and $e_{LMS}(k)$ (Figure 1) whereas the time–varying parameter $\lambda(k)$ is updated based on the overall output error $e(k)$. This gives good balance to the algorithm and the unwanted error feedback effects are avoided.

Parameter $\lambda(k)$ is updated using a stochastic gradient adaptation, given by

$$\lambda(k+1) = \lambda(k) - \mu_a \nabla_\lambda E(k)_{|\lambda=\lambda(k)} \tag{8}$$

where μ_a is a small adaptation step size. From (7) and (8) the λ update can be evaluated as

$$\begin{aligned} \lambda(k+1) &= \lambda(k) - \frac{\mu_a}{2}\frac{\partial e^2(k)}{\partial \lambda(k)} \\ &= \lambda(k) + \mu_a e(k)\big(y_{GNGD}(k) - y_{LMS}(k)\big) \end{aligned} \tag{9}$$

The overall weight vector of the convex combination of GNGD and LMS can be expressed as

$$\mathbf{w}(k) = \lambda(k)\mathbf{w}_{GNGD}(k) + (1-\lambda(k))\,\mathbf{w}_{LMS}(k) \tag{10}$$

Notice some functional similarity with the mixed norm approach [4].

To preserve the convexity[1] of this combination of adaptive filters, we need to ensure that the values of parameter $\lambda(k)$ remain within the range $0 < \lambda(k) < 1$. To that cause, in [6] a squashing sigmoid function was used as a post–nonlinearity to bound updates of λ. In our case, for a reasonably small μ_a this was not necessary and the values of λ stayed within the required bounds $0 < \lambda < 1$.

The operation ot the hybrid filter is based on the collaboration of the constitutive adaptive filters, and its computational complexity is a combination of the computaional complexity of the individual filters and the complexity of the update of the mixing paremeter. More specifically, this is found to be $\mathcal{O}(N)$ or 7N+9 multiplications and 7N+1 addtions per iteration.

4. SIMULATIONS

The performance of the proposed approach was evaluated in the prediction setting, and was compared to that of LMS and GNGD. The length of all the adaptive filters considered was set to $N = 10$. Convergence curves were averaged over a set of 1000 independent simulation runs.

For generality, the inputs used in simulations were:-

1. A stable linear AR(4) process given by

$$\begin{aligned} x(k) = 1.79\,x(k-1) - 1.85\,x(k-2) + 1.27\,x(k-3) \\ - 0.41\,x(k-4) + n(k) \end{aligned} \tag{11}$$

2. A benchmark nonlinear signal given by [11]

$$x(k+1) = \frac{x(k)}{1+x^2(k-1)} + n^3(k) \tag{12}$$

where $n(k)$ is a zero mean and unit variance white Gaussian process.

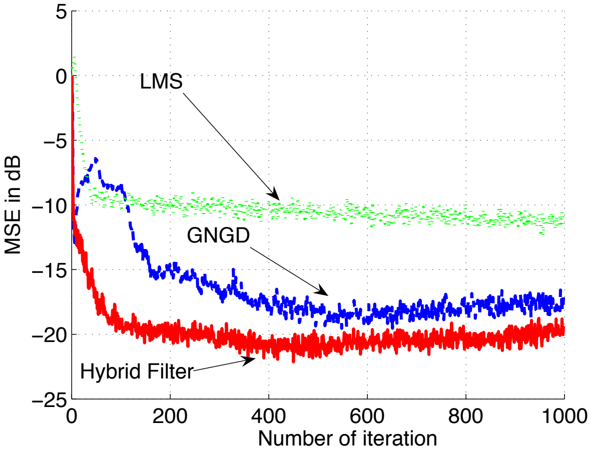

Fig. 3. Averaged performance of the GNGD, LMS and Hybrid Filter in a prediction setting for linear signal (11).

In the case of linear signal (11), the values of parameters used within GNGD were $\mu_{GNGD} = 1.95$ and $\rho = 0.15$. The initial value for the regularisation parameter was $\varepsilon(0) = 0.1$. To achieve good steady state performance, the step–size of LMS was relatively small, and was chosen to be $\mu_{LMS} = 0.01$. Within the convex combination of filters trained by GNGD and LMS, $\mu_\alpha = 0.05$ was employed as a value of the step–size for the adaptation of $\lambda(k)$. From the convergence curves shown in Fig. 3, the proposed convex combination of filters outperformed both the single LMS and GNGD. The proposed approach achieved as fast convergence as GNGD in the beginning of adaptation and eventually attained as good steady state performance as that of the LMS algorithm (after 10000 samples – not shown in Figure 3).

The results of the same experiment conducted on a nonlinear signal (12) are shown in Figure 4. The parameters of the GNGD algorithm were set to $\mu_{GNGD} = 0.6$, $\rho = 0.15$ and $\varepsilon(0) = 0.1$. The step size of the LMS was $\mu_{LMS} = 0.001$ and $\mu_\alpha = 0.05$ was chosen for the step size within the update of $\lambda(k)$. Clearly, the proposed convex approach was able to combine successfully the small settling time of GNGD with

[1] Let points x and y lie on a line. Then a convex combination $\lambda x + (1 - \lambda)y$ will lie in between x and y on the same line for $\lambda \in (0, 1)$.

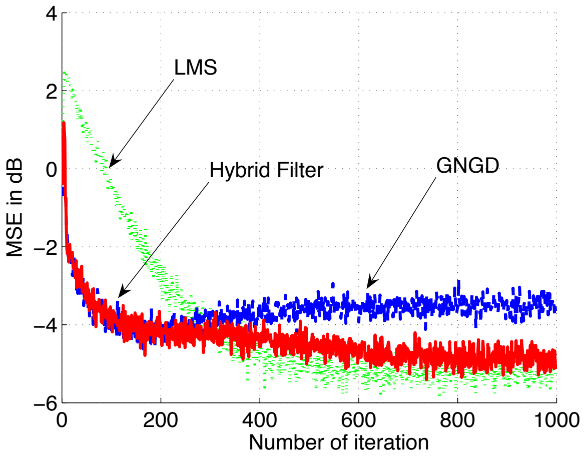

Fig. 4. Averaged performance of GNGD, LMS and Hybrid Filter for non-linear signal (12).

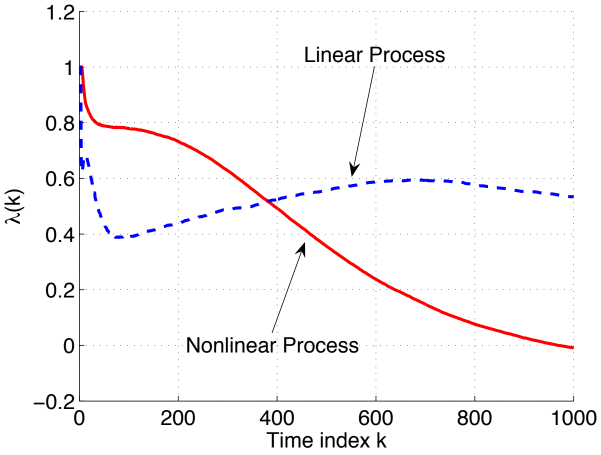

Fig. 5. Time evolution of the adaptive convex parameter $\lambda(k)$ for the linear and the nonlinear case.

the low steady state MSE of LMS and hence outperformed its component filters. The benchmark nonlinear signal (12) used in the simulations is extremely complex and difficult to predict, which illustrates the validity of the proposed approach. As desired, the convergence curve of the hybrid filter followed closely that of GNGD in the beginning of adaptation and then continued to follow the convergence curve of LMS.

Since the proposed algorithm relies on GNGD for rapid initial convergence, and then on LMS to preserve small steady state misalignment, it is clear that the former must have a relatively large adaptation parameter ρ whereas the latter should use a very small step size. The value of parameter $\lambda(k)$ was therefore initially set to unity (output $y(k)$ dominated by y_{GNGD}) and it is expected that after convergence $\lambda(k) \approx 0$. This is illustrated in Figure 5 for both the linear and nonlinear case. In addition, our experiments show that:-

- if one of the constitutive filters fails to converge then the values of $\lambda(k)$ are such that the hybrid filter follows the stable subfilter;
- if both the constitutive filters achieve considerable values of steady state error, $\lambda(k)$ converges to a value $\lambda_\infty \in (0, 1)$ which is significantly far from zero.

5. CONCLUSIONS

To improve the steady–state performance of the Generalized Normalized Gradient Descent (GNGD) algorithm, we have employed a convex combination of two adaptive filters in which one filter is trained by GNGD and the other by the Least Mean Square (LMS). This hybrid filter has been shown to have as good a steady-state performance as LMS, whilst keeping the fast convergence and good tracking capabilities of GNGD. This convex combination of GNGD and LMS allows the hybrid filter to remain fast responding in a nonstationary environment and to settle in a stationary environment.

6. REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, third edition, 1996.

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

[3] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximation*, Springer–Verlag: New York, 1990.

[4] J. Chambers and A. Avlonitis, “A robust mixed-norm adaptive filter algorithm,” *IEEE Signal Processing Letters*, vol. 4, no. 2, pp. 46–48, 1997.

[5] D. P. Mandic, “A general normalised gradient descent algorithm,” *IEEE Signal Processing Letters*, vol. 11, no. 2, pp. 115–118, 2004.

[6] A. R. Figueras-Vidal, J. Arenas-Garcia, and A. H. Sayed, “Steady state performance of convex combinations of adaptive filters,” in *Proceedings of the International Conference on Acoustis, Speech and Signal Processing (ICASSP’05)*, pp. 33–36, 2005.

[7] 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.

[8] 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.

[9] D. P. Mandic, D. Obradovic, and A. Kuh, “A robust general normalised gradient descent algorithm,” in *Proceedings of IEEE SPS Workshop*, pp. 133–136, 2005.

[10] C. Darken and J. Moody, “Towards faster stochastic gradient search,” in *Neural Information Processing Systems 4*, J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds., pp. 1009–1016. Morgan Kaufmann, 1992.

[11] 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.