EVALUATION AND DESIGN OF VARIABLE STEP SIZE ADAPTIVE ALGORITHMS

Cássio Guimarães Lopes [1], *José Carlos Moreira Bermudez* [2*]

Federal University of Santa Catarina
Department of Electrical Engineering
Campus Universitário, Trindade – Fpolis, SC
[1] cassio@eel.ufsc.br [2] bermudez@fastlane.com.br

ABSTRACT

This paper presents a new methodology for evaluation and design of variable step size adaptive algorithms. The new methodology is based on a learning plane, which combines the evolutions of both the step size and the mean square error. It includes both transient and steady-state behaviors and can be used to compare performances of different algorithms against an optimum trajectory in the learning plane. The new technique can also be used for algorithm optimization in system identification applications.

1. INTRODUCTION

Variable step size adaptive algorithms are useful in several practical applications [1], [5], [6], [7], [4]. A proper step size adjustment strategy can improve algorithm's performance (as compared to its fixed step size counterpart) during both the transient and steady-state (including tracking in nonstationary environments) phases of adaptation [5]. Several step size control approaches have been proposed in the literature [6], [7], [8], [9], [10], [11], [12], [13], [14]. Most include parameters which control the time evolution of the step size. Thus, using variable step size algorithms raises two questions: (i) What is the best step size sequence for a given application, and (ii) How to adjust the algorithm's parameters to generate this sequence.

Most design equations available for variable step size algorithms are based on steady-state objectives. The algorithm's transient behavior is usually verified through simulations. Different algorithms are compared by confronting their simulated transient performances for a given steady-state behavior. It would be desirable to consider both transient and steady-state performances in an algorithm's design procedure. It would also be important to benchmark a given design against a theoretical optimum performance, not just against the performance of another algorithm which may have not been designed for optimal performance.

This paper proposes a new performance evaluation methodology. It is based on a theoretic optimal step size sequence and on an adequate simultaneous representation for both the cost function and the step size time evolutions. Transient and steady-state peformance objectives can be benchmarked against the optimum algorithm behavior. The new methodology can also be used for algorithm design optimization in the important case of system identification with a white reference input. The proposed technique is applied to the design of the simple VSS algorithm proposed in [6]. It is shown that by using the proposed transient performance optimization in addition to the steady-state design equations in [6], the VSS algorithm can outperform recently proposed algorithms [7], [12].

2. OPTIMAL STEP SIZE SEQUENCE

The system in Fig. 1 can model several adaptive filtering applications [1]. $W^o = \left[w_0^o,\, w_1^o,\, \cdots,\, w_{N-1}^o \right]^T$ is an unknown system response, $W(n) = \left[w_0(n),\, w_1(n),\, \cdots,\, w_{N-1}(n) \right]^T$ is the adaptive filter response, $x(n)$ is the reference signal,

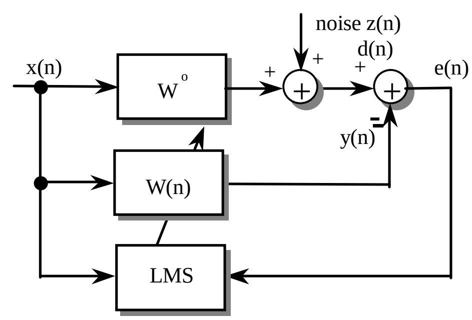

Fig.1 – Adaptive system

$X(n) = \left[x(n), x(n-1), \cdots, x(n-N+1) \right]^T$ is the input observation vector, $y(n) = X^T(n) W(n)$ is the adaptive filter output, $z(n)$ is the measurement noise which is assumed white and Gaussian (WGN), with variance σ_z^2 and uncorrelated with any other signal, $d(n) = X^T(n) W^o + z(n)$ is the noisy primary signal, and $e(n) = d(n) - y(n)$ is the estimation error. The cost function is the mean square error (MSE) $\xi(n)=E\{e^2(n)\}$. The weight update equation for the variable step size LMS algorithm is given by

$$W(n+1) = W(n) + \mu(n)\, e(n) X(n) \qquad (1)$$

where the step size $\mu(n)$ must be constrained to an interval $[\mu_{min}, \mu_{max}]$ to preserve algorithm's stability and ability to track nonstationary environments [1], [6], [7].

For $x(n)$ zero-mean Gaussian, a good estimate for the optimal step size sequence can be obtained from the well known analytical model for the fixed step size LMS algorithm [1], [2]:

* This work was supported in part by Capes (Brazilian Education Ministry) and by CNPq (Brazilian Council on Science an Technology) – Grant 352084/92-8

$$\xi(n) = \xi_{min} + tr\{RK(n)\} \qquad (2)$$

$$\begin{aligned} K(n+1) = K(n) - \mu \cdot [RK(n) + K(n)R] \\ + \mu^2 \cdot \left[R \cdot tr\{RK(n)\} + 2RK(n)R + R \cdot \xi_{min} \right] \end{aligned} \qquad (3)$$

where $R = E\{X(n)X^T(n)\}$ is the input autocorrelation matrix, $K(n) = E\{V(n)V^T(n)\}$ is the weight error vector correlation matrix ($V(n) = W(n) - W^o$), ξ_{min} is the minimum MSE and $tr\{\bullet\}$ stands for the trace of a matrix. The optimal step size sequence for stationary environments can be obtained from (2) and (3) by minimizing $\xi(n+1)$ for a given $\xi(n)$ as a function of μ [3], [12]. The optimum step size sequence is given by

$$\mu_{opt}(n) = \frac{tr\{R^2K(n)\}}{tr\{R^2\}\, tr\{RK(n)\} + 2\, tr\{R^3K(n)\} + \xi_{min} \cdot tr\{R^2\}} \qquad (4)$$

This important result is frequently neglected in the literature. Comparisons of different algorithms are usually based on simulation results [6], [7], [11], [13], [14]. The next sections will show that the results of such comparisons can be misleading.

3. THE LEARNING PLANE

Evaluation of a variable step size algorithm using separate curves for $\mu(n)$ and $\xi(n)$ is not trivial [6], [7], [11], [14]. Fig. 2 illustrates how $\mu(n)$ and $\xi(n)$ can be seen as two possible two-dimensional projections of the three-dimensional trajectory $\xi(\mu,n)$. Notice also that a third projection on the plane $[\xi,\mu]$ combines simultaneous (same index n) informations about $\mu(n)$ and $\xi(n)$ in a single curve which establishes a direct cause-effect relationship. This *learning plane* permits algorithm evaluation in the transient phase and in steady-state.

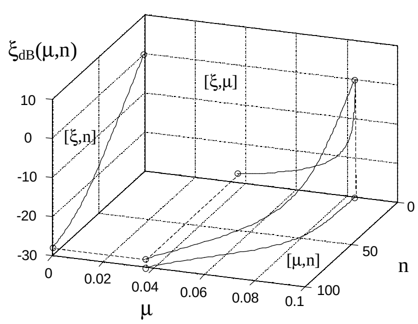

Fig. 2 – Projections of the three-dimensional trajectory

Fig. 3 shows a typical learning plane μ vs ξ. The axes are reversed (compared to Fig. 2) for better visualization. Each of the regions and curves in this plane are described in the following.

3.1. The optimum trajectory

The optimum trajectory is defined by the curve obtained from the recursive equations (2), (3) and (4) for each n. The closer an adaptive algorithm's trajectory is to the optimum, the better is the algorithm's dynamic behavior.

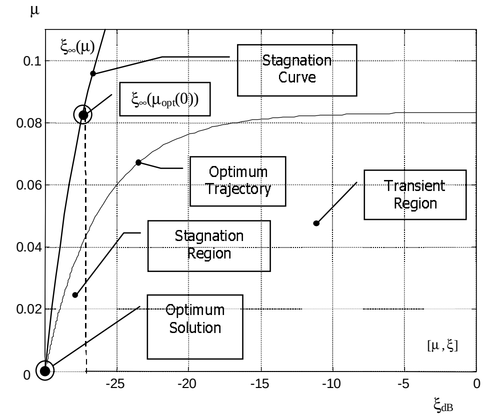

Fig.3 – The learning plane

3.2. The adaptive algorithm's learning trajectory

A variable step size algorithm is characterized by its step size update equation. Define the adaptive algorithm's learning trajectory as the sequence of mean step sizes $E[\mu(n)]$ as a function of $\xi(n)$. To obtain an analytical model for the adaptive algorithm's trajectory, the following simplifying assumptions are used [1], [6], [7], [11]:

A.1) $X(n)$ and $V(n)$ are statistically independent. This is the independence assumption, frequently used in adaptive algorithm behavior analyses [1], [5], [6], [11], [12].

A.2) $\mu(n)$ and $\mu^2(n)$ are independent of $X(n)$ and $V(n)$ [6], [7], [11], [12].

A.3) $E[\mu^2(n)] \approx E^2[\mu(n)]$.

A.2 and A.3 are accurate when the statistical fluctuations of $\mu(n)$ are small when compared to those of $X(n)$ and $e(n)$ [6], [7], [11]. Using A.1-A.3 and taking the expectation of (3) (with μ replaced with $\mu(n)$) over $\mu(n)$ yields

$$\begin{aligned} K(n+1) = K(n) - E[\mu(n)] \cdot [RK(n) + K(n)R] \\ + E^2[\mu(n)] \cdot \left[R \cdot tr\{RK(n)\} + 2RK(n)R + R \cdot \xi_{min} \right] \end{aligned} \qquad (5)$$

Equations (2) and (5) can be combined with an analytical model for $E[\mu(n)]$ to describe the learning trajectory on the $[\mu,\xi]$ plane.

3.3. Steady-state and the stagnation curve

The stagnation curve is defined as the locus in the $[\mu,\xi]$ plane corresponding to convergence rate equal to zero, i.e.

$$\xi_\infty(\mu) = \lim_{n\to\infty} \xi(n,\mu) \qquad (6)$$

The steady-state behavior corresponds to a single point on the stagnation curve. The algorithm's trajectory starts at the extreme right and progresses with time towards the stagnation curve. It stops (steady-state) when it touches the stagnation curve.

$\xi_\infty(\mu)$ can be determined from A.1-A.3 and the equation for LMS misadjustment [1]. For $x(n)$ white [1], [2],

$$\xi_\infty(\mu) = \frac{1}{1 - \frac{N}{2}\left(\frac{\mu\,\sigma_x^2}{1-\mu\,\sigma_x^2}\right)} \cdot \xi_{min} \qquad (7)$$

where σ_x^2 is the variance of $x(n)$.

3.4. The transient and stagnation regions

The convergence rate reduces significantly as the algorithm approaches the stagnation curve. Deviations from the optimum trajectory in this region have great impact on algorithm performance. Define the boundary between a *transient region* and a *stagnation region* as the vertical line through the point $[\mu_{opt}(0),\xi(\infty)]$ in the learning plane (see Fig. 3). This point corresponds to the steady-state MSE for the fixed step size LMS algorithm with $\mu = \mu_{opt}(0)$. Experience shows that agreement between algorithm and optimum trajectories within the *stagnation region* is most important for algorithm's peformance.

3.5. The optimum solution

This is the optimum stagnation point (see Fig. 3). At this point, the steady-state mean weight vector is such that $\xi_\infty = \xi_{min}$.

4. APPLICATIONS

The initialization and the limits $[\mu_{min}, \mu_{max}]$ play important roles in the step size adaptation. A good strategy for fast convergence is to initialize the algorithm with $\mu_o=\mu_{opt}(0)$ obtained from (4) using estimates about the signal environment. For white $x(n)$ white, it can be easily shown that

$$\mu_o = \frac{\sigma_d^2 - \sigma_z^2}{(N+2)\,\sigma_x^2(\sigma_d^2 - \sigma_z^2) + N\sigma_x^2\sigma_z^2} \qquad (8)$$

where σ_d^2 is the variance of d(n). σ_d^2, σ_x^2 and σ_z^2 can be estimated from data measurements. μ_o in (8) is used in the following system identification examples. This is the most used application to compare different variable step size algorithms and allows to illustrate the use of the learning plane as a design tool.

Consider the *Variable Step Size LMS Algorithm* (VSS) proposed in [6]. Its μ adaptation equation is

$$\mu(n+1) = \alpha \cdot \mu(n) + \gamma \cdot e^2(n) \qquad (9)$$

Taking the expected value of (9) and using the expression derived in [6, Eq. 43] for steady-state misadjustment M yields

$$E[\mu(n+1)] = \alpha \cdot E[\mu(n)] + \gamma \cdot \xi(n) \qquad (10)$$

$$\gamma = \left[1 - \left(\frac{1-M}{1+M}\right)^2\right] \cdot \frac{(1-\alpha^2)}{2\,(3-\alpha)Tr\{R\}\sigma_z^2} \qquad (11)$$

Given M and an initial estimate α_o for α, $\gamma = \gamma_o$ is determined from (11). The pair $\{\alpha_o, \gamma_o\}$ leads to a trajectory in the plane $[\mu, \xi]$, which is determined from (2), (5) and (10). The initial value α_o can then be iteratively adjusted to determine a final corrected pair $\{\alpha_{cor}, \gamma_{cor}\}$ which leads to the greatest proximity between the algorithm's and the optimum trajectories. Usually, $\{\alpha_{cor}, \gamma_{cor}\}$ does not match the design proposed in [6]. Experience shows that the pair $\{\alpha_{cor}, \gamma_{cor}\}$ can be usually obtained after few (two or three) parameter adjustments, making the design optimization very simple.

4.1. Algorithm comparisons

A *Modified VSS* algorithm (MVSS) has been recently proposed in [7]. Results in [7] indicate that the MVSS algorithm outperforms the VSS algorithm. Examples presented in [7] are used here to illustrate the VSS algorithm design using the new methodology. Though the responses W^o were not provided in [7], equivalent responses can be used for the case of white x(n) (symmetric MSE surface) without affecting the conclusions.

Low SNR: In [7, Ex. 1] N=4 and x(n) is WGN with $\sigma_x^2 = 1$. z(n) is also WGN, with $\sigma_z^2 = 1$. The unknown system is W^o=[2.8, 2.8, 2.8, 2.8]^T. The parameters used in [7] were α_{mvss}=0.97, $\gamma_{mvss}=10^{-3}$, β_{mvss}=0.99, α_{vss}=0.97 and $\gamma_{vss}=10^{-5}$. These parameters lead to a steady-state excess MSE $\xi_{ex} = -34\,dB$ (M≈6.9×10^{-4}). Also, $[\mu_{min}, \mu_{max}] = [5\times10^{-4}, 0.1]$. Fig. 4 shows the optimum trajectory, as well as the learning trajectories and the MSE for the original VSS and the corrected VSS designs. Note that the trajectory for the corrected VSS algorithm is much closer to the optimum trajectory, specially in the stagnation region. It was obtained for $\mu_{min} = 5\times10^{-4}$, $\mu_{max} = \mu(0) = \mu_o$= 0.163, α_{cor}=0.973 and γ_{cor} = 9.05×10^{-6}. Note from the MSE curves that the optimized VSS algorithm outperforms the MVSS algorithm.

High SNR: [7, Ex. 3] uses μ_{max}=0.1 and $\mu_{min}=5\times10^{-4}$; α_{mvss}=0.97, β_{mvss}=0.99, γ_{mvss}=1, α_{vss}=0.97 and γ_{vss}=0.02. For these parameters $\xi_{ex} \approx -60\,dB$ and M≈1.38×10^{-3}. An equivalent system is W^o=[1.58, 1.58, 1.58, 1.58]^T. x(n) and z(n) are WGN with $\sigma_x^2 = 1$ and $\sigma_z^2 = 10^{-3}$. Trajectory correction leads to α_{cor}=0.977 and γ_{cor} =0.0154, with $\mu(0)$= μ_{max} =$\mu_o \approx$ 0.167. The results are shown in Fig. 5. The improvement obtained from the redesign of the VSS algorithm is clear.

4.2. Real time optimization vs design optimization

This example compares two different approaches for μ adjustment. The μ-adjustment algorithms proposed in [1], [11], [12] perform parameter adjustment in real time. Simpler algorithms (such as VSS) require parameter optimization in the design phase. The latter require smaller computational complexity during operation. This example compares the NASS [12] (the most recently proposed from [1], [11], [12]) and the VSS algorithms. Let $W^o = \frac{1}{\sqrt{30}} \cdot [1, 1, 1, \cdots, 1]^T$, ($|W_o|$=1), N=30 and M≈1%. x(n) and z(n) are WGN with $\sigma_x^2 = 1$ and $\sigma_z^2 = 10^{-3}$. α_{cor}=0.994 and γ_{cor}=0.0039 are determined for the VSS algorithm. $\rho = \rho_\tau$= 0.002 and L_d =1 were used for the NASS algorithm since x(n) is WGN [12]. Fig. 6 shows the design and simulation results. It is clear from the MSE curves that the VSS algorithm can have a very competitive performance if optimized using the proposed technique.

4.3. Non-stationary system identification

This example repeats the previous one with an abrupt change from W^o to $-W^o$ at n=1000 [7], [11]. Fig. 7 shows the simulation results. It can be again verified that even designed for a small misadjustment, the optimized VSS algorithm performs very well during transient after abrupt nonstationarities.

5. CONCLUSIONS

It is important to devise analytical tools for the systematic evaluation of variable step size adaptive algorithms. The new methodology proposed in this paper provides a simple and effective technique to compare algorithms derived from the LMS algorithm. It considers both transient and steady-state behaviors. In system identification, it is also possible to use the methodology for design. The results have shown that previous algorithm comparison results were based on poor designs and must be re-addressed.

6. REFERENCES

[1] Haykin, S. *Adaptive Filter Theory*. 3rd. Ed., Prentice Hall, 1996.

[2] Horowitz, L.; Senne, K. Performance Advantage of Complex LMS for Controlling Narrow-Band Adaptive Arrays. *IEEE Trans. on ASSP*, p. 722-736, June 1981.

[3] Gardner, W. A. Learning Characteristics of Stochastic-Gradient-Descent Algorithms: A General Study, Analysis, and Critique. *Signal Processing*, North Holland, v. 6, p. 113-133, 1984.

[4] Slock, D. On the Convergence Behavior of the LMS and the Normalized LMS Algorithms. *IEEE Trans. on Signal Processing*, p. 2811-2825, Sept. 1993.

[5] Evans, J. B.; Xue, P.; Liu, B. Analysis and Implementation of Variable Step Size Adaptive Algorithms. *IEEE Trans. on Signal Processing*, p. 2517-2535, Aug. 1993.

[6] Kwong, R. H.; Johnston, E. W. A Variable Step Size LMS Algorithm. *IEEE Trans. Signal Processing*, p. 1633-1642, July 1992.

[7] Abounasr, T.; Mayyas, K. A Robust Variable Step-Size LMS-Type Algorithm: Analysis and Simulations. *IEEE Trans. Signal Processing*, p 631-639, Mar. 1997.

[8] Shan, T. J.; Kailath, T. Adaptive Algorithms with an Automatic Gain Control Feature. *IEEE Trans. on Circuits and Systems*, p 122-127, Jan. 1988

[9] Okello, J.; Itoh, Y.; Fukui, Y.; Nakanishi, I. *et al.* A New Modified Variable Step Size for the LMS Algorithm. In: ISCAS-98. Monterey, CA, v. 5, pp 170-173.

[10] Kushner, H. J.; Yang, J. Analysis of Adaptive Step-Size SA Algorithms for Parameter Tracking. *IEEE Trans. Automatic Control*, p 1403-1410, Aug. 1995.

[11] Mathews, J. V.; Xie, Z. A Stochastic Gradient Adaptive Filter with Gradient Adaptive Step Size. *IEEE Trans. Signal Processing*, p 2075-2087, June 1993.

[12] Koike, S. A Novel Adaptive Step Size Control Algorithm for Adaptive Filters. In: ICASSP-99, Phoenix, Arizona, v. 4, paper no. 1054.

[13] Gan, W. S. Designing a fuzzy step size LMS algorithm, *IEE Proc. Vis. Image Signal Processing*, p. 261-266, Oct. 1997.

[14] Karni, S.; Zeng, G. A New Convergence Factor for Adaptive Filters. *IEEE Trans. on Circuits and Systems*, p. 1011-1012, July 1989.

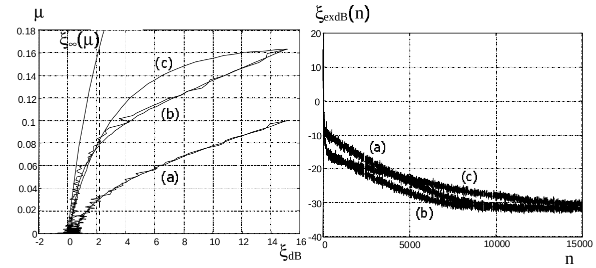

Fig. 4 – *Left*: (a,b) original and corrected VSS trajectories; (c) optimum trajectory. *Right*: (a,b) VSS original and corrected MSE; (c) MSE for the MVSS algorithm . Average of 200 realizations

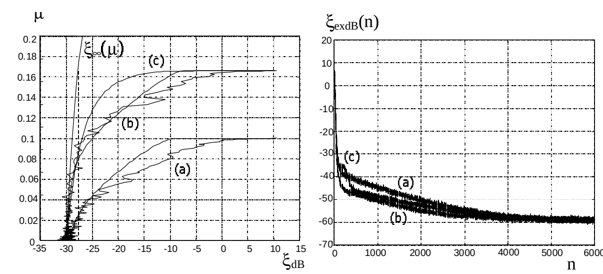

Fig. 5 – *Left*: trajectories of (a) original and (b) corrected VSS; (c) optimum trajectory. *Right:* MSE for (a) original and (b) corrected VSS; (c) MSE for the MVSS algorithm. Simulation averaging 200 runs.

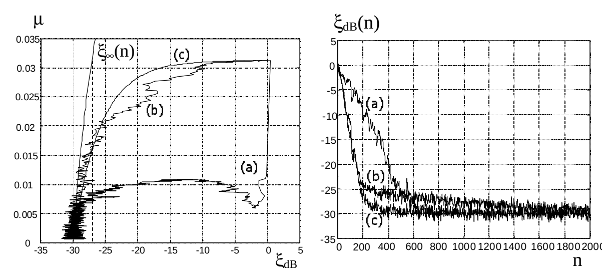

Fig. 6 – *Left:* Trajectories of (a) NASS; (b) VSS corrected; (c) optimum trajectory. *Right:* MSE of (a) NASS; (b) original VSS; (c) corrected VSS. Simulations averaging 100 runs.

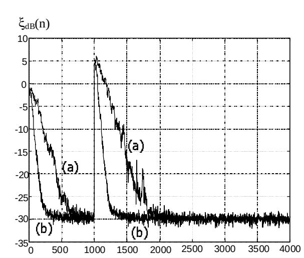

Fig. 7 – MSE in non-stationary system identification. Algorithms: (a) NASS; (b) corrected VSS. Simulation averaging 100 runs