A Comparison of Gradient-Based Algorithms for Echo Compensation with Decorrelating Properties

Markus Rupp

Institut für Netzwerk- und Signaltheorie, TH Darmstadt,
Merckstraße 25, D-6100 Darmstadt, Germany
Tel.: 6151/165406 e-mail: rupp@nesi.e-technik.th-darmstadt.de

ABSTRACT

Cancelling echoes by using the normalized least mean square (NLMS) algorithm has been state of the art for many years. In acoustical echo compensation, however, it is common to estimate more than 1000 parameters resulting in a too slow convergence when driven by speech signals. In order to overcome this drawback, a lot of modifications have been published in the last years, all having one goal: to decorrelate the driving process. Beginning with a deterministic approach we will show that all these different ideas[1, 2, 3, 7, 8, 9, 10] can be arranged in one scheme, allowing a uniform normalization. The different properties of the several algorithms are then obvious. A comparison of some algorithms with 2N-4N complexity is presented in the following paper. Surprisingly, all algorithms do not work perfectly for a big compensator filter length and speech as input process.

1. Introduction

If gradient-based algorithms are used the general adaptation rule is

$$\underline{\hat{w}}(k+1) = \underline{\hat{w}}(k) + \mu(k) e_a(k) \underline{\psi}(k) \tag{1}$$

where $\underline{\hat{w}}(k)$ are the M estimated coefficients, $\mu(k)$ is the step-size parameter, $e_a(k)$ the adaptation error and $\underline{\psi}(k)$ the gradient term, giving the direction of adaptation. If

$$e_a(k) = e(k) = d(k) - \underline{\hat{w}}^T(k)\underline{u}(k) \tag{2}$$

and $\underline{\psi}(k) = \underline{u}(k)$ is used, where $\underline{u}(k)$ is a vector with M samples of the driving process and $d(k)$ is the desired signal, the LMS algorithm is obtained. If the difference $\underline{\varepsilon}(k) = \underline{\hat{w}}(k) - \underline{w}(k)$ between the estimated parameter vector $\underline{\hat{w}}(k)$ and the searched vector $\underline{w}(k)$ is considered, a state-space approach is possible:

$$\underline{\varepsilon}(k+1) = (I - \mu(k)\underline{u}(k)\underline{u}^T(k))\underline{\varepsilon}(k) + \mu(k) e_o(k)\underline{u}(k). \tag{3}$$

The error signal can be rearranged to

$$e(k) = \underline{\varepsilon}^T(k)\underline{u}(k) - e_o(k) \tag{4}$$

with $e_o(k)$ being the additive disturbance (i.e. local speaker). For this state-space approach an eigenvalue analysis is possible. There exist $M-1$ linear independent eigenvectors orthogonal to $\underline{u}(k)$. Every corresponding eigenvalue equals one. The last eigenvector is just $\underline{u}(k)$. Its corresponding eigenvalue $\lambda(k)$ is decisive for the algorithms convergence speed. Equation (3) leads to:

$$\lambda(k) = 1 - \mu(k)\underline{u}^T(k)\underline{u}(k). \tag{5}$$

Obviously, normalization is done by choosing (NLMS algorithm)

$$\mu(k) = \frac{\alpha}{\underline{u}^T(k)\underline{u}(k)}, \tag{6}$$

resulting in a time independent eigenvalue $\lambda = 1 - \alpha$ and an optimal normalized step-size $\alpha_{opt} = 1$. As a conclusion of this analysis it can be stated that the NLMS algorithm is capable to cause fast convergence (i.e. $\lambda = 0$ for $\alpha = 1$) in one direction $\underline{u}(k)$. But for strongly correlated signals like speech this direction does not change very much, whereas for a white signal a new direction results with every new input sample. Similiar to the white process situation algorithms with decorrelating properties try to find more directions $\underline{\psi}(k)$ for every time instant k in the presence of correlated processes.

2. The Algorithms

2.1. The Concepts

The decorrelation algorithms considered here are characterized by using different values for the adaptation error $e_a(k)$ and the gradient term $\underline{\psi}(k)$. The several choices are listed in Table 1. Some algorithms use a filtered version $\underline{\psi}(k) = F[\underline{u}(k)]$ of the input vector $\underline{u}(k)$ to improve the NLMS drawback. The simplest ideas are the Ozeki-Umeda and the Mboup algorithm. The vector $\underline{\psi}(k)$ is found by either filtering the vector sequence $\underline{u}(k)$ or most simply filtering only the newest element of this vector. But exchanging only $\underline{u}(k)$ in the LMS algorithm by a new vector $\underline{\psi}(k)$ the adaptation process may become instable. In order to improve the situation Acker and Sommen used additional filtering of the adaptation error signal $e_a(k)$. But as will be shown the improvement works only for small step sizes, resulting in slow convergence. Schultheiß, however, found a possibility to overcome this drawback by filtering the desired signal $d(k)$ instead of $e(k)$.

Algorithm	$e_a(k)$	$\underline{\psi}(k)$	Orthogonality
NLMS	$e(k)$	$\underline{u}_i(k)$	no
Ozeki (P=1)	$e(k)$	$\underline{u}_i(k) - \frac{\underline{u}^T(k-1)\underline{u}(k)}{\underline{u}^T(k-1)\underline{u}(k-1)}\underline{u}_i(k-1)$	so
Mboup	$e(k)$	$F[\underline{u}_i(k)]$	co
FA	$e(k)$	$\underline{u}_i(k-i) - \frac{\underline{u}^T(k-1-i)\underline{u}(k-i)}{\underline{u}^T(k-1-i)\underline{u}(k-1-i)}\underline{u}_i(k-1-i)$	co
Acker	$F[e(k-D)]$	$F[\underline{u}_i(k)]$	co
Sommen	$F[e(k)]$	$F[\underline{u}_i(k)]$	co
Schultheiß	$F[d(k)] - \underline{\hat{w}}^T(k)F[\underline{u}(k)]$	$F[\underline{u}_i(k)]$	vo

Table 1: Choice of $e_a(k)$ and $\underline{\psi}(k)$

2.2. Normalization

In literature algorithms are usually given with fixed step sizes. For practical use, however, normalization is necessary to make the algorithms independent of changings in the input signal level. In [5, 6] it is shown that a good choice of normalization can cause an enormous improvement of the convergence speed. Therefore, a good normalization of the step-size is necessary to compare the various algorithms fairly. Since different values for adaptation error and gradient term are used, different normalizations lead to optimal eigenvalues. The normalization rules used here can be explained easily by analysing the algorithms.

3. Analysis of the Algorithms

3.1. Demands

Various algorithms use different concepts to cause decorrelation of the input process. Since $\underline{\psi}(k)$ gives the direction of the update, it is desirable to have a new direction with every step. This can be achieved by the following demands:

1. strict orthogonality (so):

$$\underline{\psi}^T(k)\underline{\psi}(k-j) = 0 \qquad \text{;for } j = 1..M,$$

2. vector orthogonality in the mean (vo):

$$\mathrm{E}[\underline{\psi}^T(k)\underline{\psi}(k-j)] = 0 \qquad \text{;for } j = 1..M,$$

3. components orthogonality in the mean (co):

$$\mathrm{E}[\underline{\psi}_i^T(k)\underline{\psi}_j(k-j)] = 0 \qquad \text{;for } i = 1..M, j = 1..M.$$

In principle, every one of these demands can be used to calculate a new direction $\underline{\psi}(k)$. Demand 1 leads to one unique solution. For $j = 1$ Ozeki-Umeda's AP algorithm is obtained. This concept can be expanded to $j = 1, 2, .., p, .., M-1$, resulting in faster algorithms for AR(p) processes. The drawback, however, is the increasing computational complexity of order $(2+p)M$. Although various decorrelation procedures can be applied for items 2 and 3, the authors used either Levinson-Durbin [1, 7] or an NLMS algorithm as predictor [2, 8, 9]. When Levinson-Durbin algorithm is used, a predictor order of $P = 8$ is typical. Since speech signals change their characteristics only after 10-20ms the decorrelating procedures do not have to be applied very often saving computational load. But every time the predictor filter coefficients have been calculated anew, the whole vector $\underline{\psi}(k)$ has to be recalculated, preventing transient effects. Since this is not done very often the effort is low in comparison to M. The NLMS algorithm, however, is used every step for investigating the predictor coefficients. Since the coefficients do not change very fast from step to step, the transient effects are low and therefore, recalculating of $\underline{\psi}(k)$ is not necessary. In principle, it is possible to exchange the decorrelation part of the several procedures resulting in new derivatives of the algorithms.

3.2. Classification

The algorithms can be grouped into three classes:

1. Only the gradient term $\underline{\psi}(k)$ is chosen different to the LMS algorithm: [2, 3, 10], leading to the following state-space approach similar to (3):

$$\underline{\varepsilon}(k+1) = (I - \mu(k)\underline{\psi}(k)\underline{u}^T(k))\underline{\varepsilon}(k) + \mu(k)e_a(k)\underline{\psi}(k). \quad (7)$$

If it is assumed that the vectors $\underline{u}(k)$ and $\underline{\psi}(k)$ are not orthogonal there are $M-1$ eigenvectors orthogonal to $\underline{u}(k)$ corresponding to the eigenvalue one. The decisive eigenvalue is:

$$\lambda(k) = 1 - \mu(k)\underline{u}^T(k)\underline{\psi}(k). \quad (8)$$

Two normalization rules can be applied [6]:

(a) Normalization Rule 1:

$$\mu(k) = \frac{\alpha}{\underline{u}^T(k)\underline{\psi}(k)}$$

(b) Normalization Rule 2:

$$\mu(k) = \frac{\alpha \underline{u}^T(k)\underline{\psi}(k)}{\underline{u}^T(k)\underline{u}(k)\underline{\psi}^T(k)\underline{\psi}(k)}$$

Both normalizations set the decisive eigenvalue into the unit circle for $\alpha \in [0, 2]$. Rule 1 has the advantage of fast

convergence of the homogenious system in (7) but the disturbence term can be increased dramatically if $\underline{u}(k)$ and $\underline{\psi}(k)$ are close to be orthogonal. Here, Rule 2 is better but may result in lower convergence speed. If the state-space approach in (7) is considered as a mapping operation, the whole mapping, however, is not necessarily contracting!

2. Additional to $\underline{\psi}(k)$ the adaptation error is filtered as in [1, 8, 9]:

$$e_a(k) = F[e(k)] = e(k) + \sum_{i=1}^{P} f_i\, e(k-i)\,. \qquad (9)$$

Here, the filter coefficients f_i are fixed, but in order to be able to track the changing correlation properties of the input sequence, the filter coefficients have to change slowly in time. Equation (9) leads to a state-space approach with a difference vector equation of higher order:

$$\begin{aligned}\underline{\varepsilon}(k+1) &= \underline{\varepsilon}(k) - \mu(k)\underline{\psi}(k)\underline{u}^T(k)\underline{\varepsilon}(k) \qquad (10)\\ &- \mu(k)\underline{\psi}(k)\sum_{i=1}^{P} f_i\, \underline{u}^T(k-i)\underline{\varepsilon}(k-i)\\ &+ \mu(k)\underline{\psi}(k)\left(e_o(k) + \sum_{i=1}^{P} f_i\, e_o(k-i)\right).\end{aligned}$$

Typically, smaller step sizes are necessary to assure convergence[4] and therefore, usually worse dynamic behavior occurs. In the next step the homogeneous solution of the case ($P = 1$) for a system of order one is investigated:

$$\begin{bmatrix} \underline{\varepsilon}(k+1) \\ \underline{\varepsilon}(k) \end{bmatrix} = \begin{bmatrix} \underline{I} - \mu\underline{\psi}(k)\underline{u}^T(k) & -f_1\,\mu\underline{\psi}(k)\underline{u}^T(k-1) \\ \underline{I} & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{\varepsilon}(k) \\ \underline{\varepsilon}(k-1) \end{bmatrix}.$$

For small step sizes it can be assumed that $\underline{\varepsilon}(k)$ does not change very quickly, and the resulting difference equation is of first order again. The optimal normalization for this case reads:

$$\begin{aligned}\mu(k) &= \frac{\alpha}{F[\underline{u}^T(k)]F[\underline{u}(k)]}\\ &= \frac{\alpha}{\underline{\psi}^T(k)\underline{\psi}(k)}. \qquad (11)\end{aligned}$$

This rule has been used successfully for the two algorithms. Using this normalization the only eigenvalues unequal to one or zero are:

$$\lambda_{1,2} = \frac{1-\alpha}{2} \pm \sqrt{\left(\frac{1-\alpha}{2}\right)^2 - f_1\alpha\frac{\underline{u}^T(k-1)\underline{\psi}(k)}{\underline{\psi}^T(k)\underline{\psi}(k)}}\,. \qquad (12)$$

Obviously, the eigenvalues are still varying with time depending on the incomming data. Let be:

$$c_1 = f_1\frac{\underline{u}^T(k-1)\underline{\psi}(k)}{\underline{\psi}^T(k)\underline{\psi}(k)}. \qquad (13)$$

The coefficient c_1 varies with time, also depending on the correlation of the input process. If the predictor coefficient f_1 is computed as in the AP algorithm, $c_1 = 0$. Figure 1 depicts the situation.

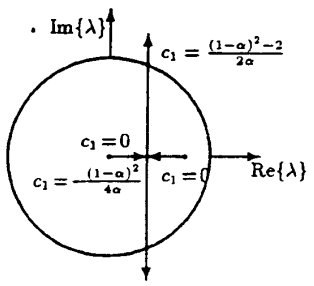

Figure 1: Eigenvalues $\lambda_{1,2} = \frac{1-\alpha}{2} \pm \sqrt{\left(\frac{1-\alpha}{2}\right)^2 - c_1\alpha}$ as a function of c_1

3. Additional to $\underline{u}(k)$ the desired $d(k)$ is also filtered [7] by F resulting to the following state-space equation:

$$\underline{\varepsilon}(k+1) = (I - \mu(k)\underline{\psi}(k)\underline{\psi}^T(k))\underline{\varepsilon}(k) + \mu(k)e_o(k)\underline{\psi}(k)\,. \qquad (14)$$

Obviously, the system is again of order one, and moreover, symmetrically as the NLMS algorithm. Therefore, the simple Normalization Rule 1:

$$\mu(k) = \frac{\alpha}{\underline{\psi}^T(k)\underline{\psi}(k)} \qquad (15)$$

achieves a contracting mapping and, thus, a convergent algorithm.

4. Simulation Results

Simulation results of the considered algorithms are given next. A measured room impulse-response of $M = 1024$ coefficients has been used. Speech samples as well as a corresponding AR-SIRP[4, 5] have been applied as driving processes. Since a speech signal allows to compute only one sample function, averaging with the artificial SIRP gives stronger results. The first column gives the averaged ERLE of 10 random runs using the AR process, the second column the resulting ERLE using a speech sample function (SF). All algorithms has been implemented with the decorrelation parts as decribed in the referenced papers. Other combinations have been checked, but no result is worth mentioning. FA is a new simple filter algorithm with very low complexity. It works like Mboub's algorithm but with only one decorrelation coefficient calculated as in Ozeki-Umeda's AP algorithm.

Since every algorithm has some freedom in the choice of its parameters, they have been chosen to cause highest convergence speed for the given situation. Because of the high computational load all algorithms have first been checked with a

Algorithm	ERLE (AR) in dB	ERLE (SF) in dB	Complexity
NLMS (white)	212	212	theoretical limit
NLMS	25	33	2M
Ozeki (P=1)	42	38	3M
Ozeki (P=2)	52	45	4M
FA	38	42	2M
Mboup	25	15	2M
Acker	19	24	2M
Sommen	38	23	2M
Schultheiß	38	37	3M

Table 2: Simulation results after 50000 iterations

filter order $M = 50$. Here, almost every algorithm shows the same relative good behavior. But for larger filter lengths the differences become evident. For all algorithms a clear improvement over the NLMS algorithm has been expected but amazingly, has not been occured. The explanation for this behavior depends on the class the several algorithms belong to. The Mboub algorithm belongs to class 1. Because the direction $\underline{\psi}(k)$ and the input vector $\underline{u}(k)$ are different, Normalization Rule 2 has to be chosen to prevent increasing error terms. Although, there is an improvement in the choice of the direction $\underline{\psi}(k)$, the corresponding eigenvalue cannot be chosen minimal causing a slower convergence. The remaining algorithms [1, 8, 9] use filtered errors and therefore, small step sizes ($\alpha = 0.3$) to assure convergence. Only the algorithms of Ozeki-Umeda and Schultheiß allow an optimal step-size and therefore fastest convergence. For both algorithms, Normalization Rule 1 and 2 are identical. Surprisingly, although the FA algorithm with Normalization Rule 2 ($\alpha = 1$) does not show good behavior for small compensator filter lengths, it behaves as well as other algorithms for long filters and speech signals. In order to improve the situation a much larger predictor length has to be chosen. The AR process used here to describe the speech signal has an order of $P = 77$. But the computational load for the various algorithms would be increased a lot when applying this predictor filter length.

5. Conclusion

As a conclusion it can be stated that all algorithms with decorrelating properties did not work as well as expected for speech signals. Ozeki-Umeda's AP algorithmus as well as Schultheiß algorithmus show very good theoretical properties resulting in optimal behavior. In spite of the improvements there is still a big gap in comparison to the NLMS algorithm with white excitation. Surprisingly, a very simple algorithm with only a 2N-complexity behaves very well for long filter lengths and speech signals.

References

1. Acker, C., Vary, P., Ostendarp, H. "Acoustic echo cancellation using prediction residual signals," *Proc. of the Second European Conf. on Speech Communication and Technology*, Genoya, Italy, 1991, pp. 1297-1300.
2. Mboup, M., Bonnet, M., "Une nouvelle structure blanchissante pour annulation d'echo acoustique," *Treizieme Colloque Gretsi*, Juan-Les-Pin du 16 au 20 Septembre, 1991, in French.
3. Ozeki, K., Umeda, T., "An adaptive filtering algorithm using orthogonal projection to an affine subspace and its properties," *Electronics and Communications in Japan*, Vol. 67-A, no. 5, 1984, pp. 19-27.
4. Rupp, M., "Über die Analyse von Gradientenverfahren zur Echokompensation," *Fortschrittberichte VDI*, Reihe 10, Nr. 242, 1993, in German.
5. Rupp, M., "The behavior of LMS and NLMS algorithms in the presence of spherically invariant processes," *IEEE Trans. Sig. Proc.*, Vol. 41, no. 3, 1993, pp. 1149-1161.
6. Rupp, M., "Normalization and convergence of adaptive IIR filters," *Proc. ICASSP*, Minneapolis, 1993.
7. Schultheiß, U., "Über die Adaption eines Kompensators für akustische Echos," *Fortschrittberichte VDI*, Reihe 10, Nr. 90, 1988, in German.
8. Sommen, P.C.W., van Valburg, C. J., "Efficient realisation of adaptive filter using an orthogonal projection method," *Proc. ICASSP*, Glascow, 1989, pp. 940-943.
9. Sommen, P.C.W., "Adaptive filtering methods," PhD-Thesis, Universitäet Eindhoven, 1992.
10. Yasukawa, H., Shimada, S., Furukawa, I., "Acoustic Echo Canceller with High Speech Quality," *Proc. ICASSP*, Dallas, 1987, pp. 49.8.1-49.8.4.