ITERATIVE ALGORITHMS FOR MULTICHANNEL EQUALIZATION IN SOUND REPRODUCTION SYSTEMS

M. Guillaume, Y. Grenier, and G. Richard

École Nationale Supérieure des Télécommunications
Département TSI. 46 Rue Barrault
75634 Paris Cedex 13, France

ABSTRACT

A fast iterative algorithm, with computation based on the fast Fourier transform (FFT), is presented. It can be used to control a soundfield at several control points with a loudspeaker array from multiple reference signals. It designs an equalizer able to invert long FIR filters and which achieves better performance than traditionnal FFT-based deconvolution methods with an equal number of coefficients in the inverse filters.

1. INTRODUCTION

Multichannel equalization is used in sound reproduction systems such as transaural sytems, ambisonics, or WFS, in order to improve the spatio-temporal quality of sound reproduction. Miyoshi and al. first introduced the MINT theorem [1] in order to have a full control of sound at several points. But, like acoustic echo control [2], multichannel equalization is an application of very high order filters, and the resolution of the subjacent problem prevents a direct inversion as recommended by Miyoshi due to the size of the resultant matrix. Kirkeby and al. proposed a fast deconvolution method [3], able to inverse long electroacoustic responses. They use a regularization parameter, weighting the effort term of the solution, in order to limit the temporal extent of inverse filters so that linear convolutions could be correctly approximated with circular ones.

In this paper, iterative algorithms are presented for the purpose of multichannel equalization. They are able to invert long multiple electroacoustic impulse responses, and achieve better results than traditionnal methods with the same number of coefficients in the inverse filters. The global aim is to minimize a cost function expressed in the time domain but computations of the update equalizer are more efficiently done in the frequency domain, using circular convolutions (FFT) to calculate linear ones. In section 2, the multichannel equalization problem and the notations used are introduced. In section 3, two different algorithms are presented, and in section 4, results are presented in the case of the cross-talk cancellation problem and compared with other methods. Finally, section 5 suggests some conclusions.

2. CONTEXT AND OBJECTIVES

2.1. Problem description and variable definitions

Figure 1 illustrates the discrete-time multichannel equalization problem encountered in generalized transaural reproduction systems [4]. The objective of the equalization is to determine the loudspeaker array inputs $\underline{\boldsymbol{y}}(n)$ such that the sound pressure is controlled at multiple positions in space. In order to achieve this objective, I desired signals $\underline{\boldsymbol{s}}(n)$ are computed from K recorded signals $\underline{\boldsymbol{x}}(n)$ described by the extrapolation relation :

$$\begin{pmatrix} s_1(n) \\ \vdots \\ s_I(n) \end{pmatrix} = \begin{bmatrix} a_{11}(n) & \dots & a_{1K}(n) \\ \vdots & \ddots & \vdots \\ a_{I1}(n) & \dots & a_{IK}(n) \end{bmatrix} * \begin{pmatrix} x_1(n) \\ \vdots \\ x_K(n) \end{pmatrix} \quad (1)$$

When $K = 1$, $x(n)$ represents the temporal signature of a point source, whereas the extrapolation operator A characterizes the propagation in an enclosed space from the source to the I control points.

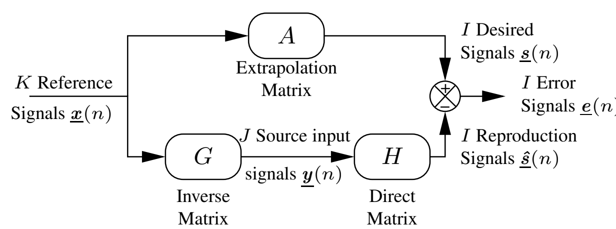

Fig. 1. Multichannel discrete-time equalization problem

The multiple input multiple output (MIMO) system H describes the electroacoustic transfer functions. It relates the J transducers signals to the I reproduced signals at the control points :

$$\begin{pmatrix} \hat{s}_1(n) \\ \vdots \\ \hat{s}_I(n) \end{pmatrix} = \begin{bmatrix} h_{11}(n) & \dots & h_{1J}(n) \\ \vdots & \ddots & \vdots \\ h_{I1}(n) & \dots & h_{IJ}(n) \end{bmatrix} * \begin{pmatrix} y_1(n) \\ \vdots \\ y_J(n) \end{pmatrix} \quad (2)$$

Thus $h_{ij}(n)$ is the electroacoustic impulse response between the transducer j and the control point i.

The MIMO system G is the equalizer. It computes the input signals of the loudspeaker array from the K reference signals :

$$\begin{pmatrix} y_1(n) \\ \vdots \\ y_J(n) \end{pmatrix} = \begin{bmatrix} g_{11}(n) & \dots & g_{1K}(n) \\ \vdots & \ddots & \vdots \\ g_{J1}(n) & \dots & g_{JK}(n) \end{bmatrix} * \begin{pmatrix} x_1(n) \\ \vdots \\ x_K(n) \end{pmatrix} \quad (3)$$

Thus $g_{jk}(n)$ is the impulse response of the equalizer between the reference signal k and the loudspeaker input signal j.

Using equation 3, 2 and 1, it is easily shown that the equalizer has to achieve $HG = A$, independently of the K reference signals

corresponding author : mathieu.guillaume@enst.fr

[4]. The problem can then be decomposed into K independent subproblems [5] :

$$\forall k \in [1,\ldots,K],\quad \begin{bmatrix} h_{11}(n) & \ldots & h_{1J}(n) \\ \vdots & \ddots & \vdots \\ h_{I1}(n) & \ldots & h_{IJ}(n) \end{bmatrix} * \begin{pmatrix} g_{1k}(n) \\ \vdots \\ g_{Jk}(n) \end{pmatrix} = \begin{pmatrix} a_{1k}(n) \\ \vdots \\ a_{Ik}(n) \end{pmatrix} \tag{4}$$

2.2. Numerical resolution

Each above subproblem is of infinite dimension and some approximations must be made to solve it numerically. Each impulse response $h_{ij}(n)$, $g_{jk}(n)$, and $a_{ik}(n)$ will be modeled by a FIR filter [2] of L_h, L_g, and $N = L_h + L_g - 1$ (the length of the discrete convolution of h_{ij} and g_{jk}) coefficients. In the following, we use the notations :

$$\underline{\boldsymbol{g}}_{jk} = [g_{jk}(0),\ldots,g_{jk}(L_g-1)]^T$$
$$\underline{\boldsymbol{a}}_{ik} = [a_{ik}(0),\ldots,a_{ik}(N-1)]^T$$
$$\underline{\boldsymbol{h}}_{ij} = [h_{ij}(0),\ldots,h_{ij}(L_h-1)]^T$$

$$\underset{[N\times L_g]}{\boldsymbol{\mathcal{H}}_{ij}} = \begin{bmatrix} h_{ij}(0) & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ h_{ij}(L_h-1) & & \ddots & 0 \\ 0 & \ddots & & h_{ij}(0) \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & h_{ij}(L_h-1) \end{bmatrix} \tag{5}$$

$$\boldsymbol{\mathcal{H}} = \begin{bmatrix} \boldsymbol{\mathcal{H}}_{11} & \ldots & \boldsymbol{\mathcal{H}}_{1J} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\mathcal{H}}_{I1} & \ldots & \boldsymbol{\mathcal{H}}_{IJ} \end{bmatrix}$$

$$\underline{\boldsymbol{g}}_k = \left[\underline{\boldsymbol{g}}_{1k}^T,\ldots,\underline{\boldsymbol{g}}_{Jk}^T\right]^T;\ \underline{\boldsymbol{a}}_k = \left[\underline{\boldsymbol{a}}_{1k}^T,\ldots,\underline{\boldsymbol{a}}_{Ik}^T\right]^T$$

Thus, $\boldsymbol{\mathcal{H}}_{ij}\cdot\underline{\boldsymbol{g}}_{jk}$ computes the discrete convolution between the two finite sequences h_{ij} and g_{jk}. The resulting problem is of finite dimension and can be solved numerically. With the above notations, it can be synthesized by the relation :

$$\boldsymbol{\mathcal{H}}\underline{\boldsymbol{g}}_k = \underline{\boldsymbol{a}}_k,\ \forall k \in [1,\ldots,K] \tag{6}$$

Relation 6 describes a system of IN equations with JL_g unknowns. It is well known [5] that the solution which minimizes the quadratic error :

$$\underline{\boldsymbol{g}}_{k_{opt}} = \boldsymbol{\mathcal{H}}^{\dagger}\underline{\boldsymbol{a}}_k \tag{7}$$

where $\boldsymbol{\mathcal{H}}^{\dagger}$ is the pseudo-inverse of the matrix $\boldsymbol{\mathcal{H}}$. This solution is also of minimal effort in the case in which the problem is subdetermined, that is when $IN < JL_g$.Unfortunately, the practical dimensions of $\boldsymbol{\mathcal{H}}$ prevent a direct pseudo-inversion. Iterative algorithms represent an alternative solution to a direct pseudo-inversion since they converge to the optimal solution without needing the inversion of $\boldsymbol{\mathcal{H}}$.

3. ITERATIVE ALGORITHMS

At each iteration of the algorithm, the equalizer $\underline{\boldsymbol{g}}_k$ is updated in order to decrease a cost function. Generally, the mean-squared-error (MSE) is used [6] :

$$J_k\left(\underline{\boldsymbol{g}}_k\right) = \|\underline{\boldsymbol{a}}_k - \boldsymbol{\mathcal{H}}\underline{\boldsymbol{g}}_k\|_2 \tag{8}$$

In this paper, two algorithms are studied : the steepest-descent and the approximative Gauss-Newton algorithms [6]. In the following, the notation $\mathrm{diag}(\underline{\boldsymbol{m}})$ transforms the vector $\underline{\boldsymbol{m}}$ into a square matrix with its elements on the diagonal, and the notation $\mathrm{diag}_n(\boldsymbol{\mathcal{M}})$ applied to a matrix is a block matrix with n matrix $\boldsymbol{\mathcal{M}}$ on its diagonal.

The complexity of these algorithms can largely be decreased if the required computations are done in the frequency domain [7]. For each step of these algorithms, only linear convolutions of two finite discrete sequences have to be computed. This can be done more efficiently in the frequency domain using FFT algorithms. For instance, consider the convolution of $h_{ij}(n)$ and $g_{jk}(n)$, which is accomplished in the time domain by the matrix operation $\boldsymbol{\mathcal{H}}_{ij}\underline{\boldsymbol{g}}_{jk}$. This can also be done in the frequency domain by zero-padding the two sequences in order to have N elements, and by computing the circular convolution of the two sequences by FFT. The matrix operation can be written as :

$$\boldsymbol{\mathcal{H}}_{ij}\underline{\boldsymbol{g}}_{jk} = \mathcal{F}^{-1}\mathrm{diag}\left(\mathcal{F}\mathcal{M}_h\underline{\boldsymbol{h}}_{ij}\right)\mathcal{F}\mathcal{M}_g\underline{\boldsymbol{g}}_{jk}$$

where $\mathcal{F}$ $[N\times N]$ denotes the discrete Fourier transform matrix, $\mathcal{M}_h$ the zero-padding matrix for the sequences $\underline{\boldsymbol{h}}_{ij}$, and $\mathcal{M}_g$, the zero-padding matrix for the sequences $\underline{\boldsymbol{g}}_{jk}$ [7]. The following matrix identity is deduced :

$$\boldsymbol{\mathcal{H}}_{ij} = \mathcal{F}^{-1}\mathrm{diag}\left(\mathcal{F}\mathcal{M}_h\underline{\boldsymbol{h}}_{ij}\right)\mathcal{F}\mathcal{M}_g$$

This identity can be generalized to the matrix $\boldsymbol{\mathcal{H}}$:

$$\boldsymbol{\mathcal{H}} = \mathcal{F}_I^{-1}\mathbf{H}\mathcal{F}_J\mathcal{M} \tag{9}$$

with $\mathcal{F}_{\mathcal{I}} = \mathrm{diag}_I(\mathcal{F})$, $(\mathbf{H})_{i,j} = \mathrm{diag}\left(\mathcal{F}\mathcal{M}_h\underline{\boldsymbol{h}}_{ij}\right)$, $\mathcal{F}_J = \mathrm{diag}_J(\mathcal{M})$, and $\mathcal{M} = \mathrm{diag}_J(\mathcal{M}_g)$.

3.1. Steepest Descent Algorithm

In this algorithm, the equalizer $\underline{\boldsymbol{g}}_k$ is updated in the opposite direction of the gradient of the cost function[6] :

$$\begin{gathered} \underline{\boldsymbol{g}}_k(m+1) = \underline{\boldsymbol{g}}_k(m) - \tfrac{\mu}{2}\underline{\boldsymbol{\nabla}}_{\underline{\boldsymbol{g}}_k} J_k\left(\underline{\boldsymbol{g}}_k\right) \\ \text{with :} \\ \underline{\boldsymbol{\nabla}}_{\underline{\boldsymbol{g}}_k} J_k\left(\underline{\boldsymbol{g}}_k\right) = -2\boldsymbol{\mathcal{H}}^T\underline{\boldsymbol{e}}_k(m) = \boldsymbol{\mathcal{H}}^T\left(\underline{\boldsymbol{a}}_k - \boldsymbol{\mathcal{H}}\underline{\boldsymbol{g}}_k(m)\right) \end{gathered} \tag{10}$$

The iteration step must fullfill the condition $\mu < 2/\lambda_{max}$ where λ_{max} is the largest eigenvalue of the autocorrelation matrix $\boldsymbol{\mathcal{H}}^T\boldsymbol{\mathcal{H}}$ [6]. As this value is unknown, one has to proceed by successive trials in order to assure convergence of this algorithm. This procedure can be automated using an optimal step version of the algorithm, in which the step is chosen in order to minimize the cost function in the next iteration [6]. This operation is synthesized by the following equation :

$$\begin{gathered} \mu(m) = \left\{\mu/\nabla_\mu J_k\left(\underline{\boldsymbol{g}}_k\right) = 0\right\} \iff \mu(m) = \frac{\underline{\boldsymbol{e}}_k^T(m)\underline{\boldsymbol{c}}_k(m)}{\underline{\boldsymbol{c}}_k^T(m)\underline{\boldsymbol{c}}_k(m)} \\ \text{with : } \underline{\boldsymbol{c}}_k(m) = \boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{H}}^T\underline{\boldsymbol{e}}_k(m) \end{gathered} \tag{11}$$

The details of the computation are summarized in Table 1.

<table>
<tr><th colspan="2">Initialization</th></tr>
<tr><td colspan="2">Frequency Direct Matrix
$(\mathbf{H})_{i,j} = \mathrm{diag}\left(\mathcal{F}\mathcal{M}_h \underline{\boldsymbol{h}}_{ij}\right)$
Reference Vector
$\underline{\mathbf{a}}_k = \mathcal{F}_I \underline{\boldsymbol{a}}_k$</td></tr>
<tr><th colspan="2">Iteration</th></tr>
<tr><td colspan="2">Estimated vector
$\underline{\hat{\mathbf{a}}}_k(m) = \mathbf{H}\mathcal{F}_J\mathcal{M}\underline{\boldsymbol{g}}_k(m)$
Error vector
$\underline{\mathbf{e}}_k(m) = \underline{\mathbf{a}}_k(m) - \underline{\hat{\mathbf{a}}}_k(m)$
Gradient vector
$\underline{\boldsymbol{d}}_k(m) = \mathcal{M}^T\mathcal{F}_J^{-1}\mathbf{H}^H\underline{\mathbf{e}}_k(m)$</td></tr>
<tr><td>Normal Version</td><td>Optimal Step Version</td></tr>
<tr><td>Update
$\underline{\boldsymbol{g}}_k(m+1) = \underline{\boldsymbol{g}}_k(m) + \mu\underline{\boldsymbol{d}}_k(m)$</td><td>Optimal Step Vector
$\underline{\mathbf{c}}_k(m) = \mathbf{H}\mathcal{F}_J\mathcal{M}\underline{\boldsymbol{d}}_k(m)$
Iteration Step
$\mu(m) = \dfrac{\underline{\mathbf{e}}_k^H(m)\underline{\mathbf{c}}_k(m)}{\underline{\mathbf{c}}_k^H(m)\underline{\mathbf{c}}_k(m)}$
Update
$\underline{\boldsymbol{g}}_k(m+1) = \underline{\boldsymbol{g}}_k(m) + \mu(m)\underline{\boldsymbol{d}}_k(m)$</td></tr>
</table>

Table 1. Steepest Descent Algorithms

3.2. Approximated Gauss-Newton Algorithm

The opposite direction of the gradient of the cost function is not the optimal direction for updating the equalizer. To increase speed convergence, the best direction for updating is obtained by multiplying the gradient vector with the inverse of the Hessian matrix $\left(\boldsymbol{\mathcal{H}}^T\boldsymbol{\mathcal{H}}\right)^{-1}$ [6]. Admittedly, if the computation of the inverse Hessian matrix was feasible, it would be better to solve directly the problem without an iterative algorithm. But the inverse Hessian matrix can only be approximated, using the frequential Hessian matrix $\mathbf{H}^H\mathbf{H}$. Hence, the structure of the matrix $\left(\mathbf{H}^H\mathbf{H} + \lambda\mathrm{Id}\right)$, constituted of blocks of diagonal matrix, enables fast algorithms for its inversion. The parameter λ is used for regularization. Finally, the approximated temporal inverse Hessian matrix $\boldsymbol{\mathcal{W}}$ used here is :

$$\left(\boldsymbol{\mathcal{H}}^T\boldsymbol{\mathcal{H}}\right)^{-1} \approx \boldsymbol{\mathcal{W}} = \mathcal{M}^T\mathcal{F}_J^{-1}\left(\mathbf{H}^H\mathbf{H} + \lambda\mathrm{Id}\right)^{-1}\mathcal{F}_J\mathcal{M} \quad (12)$$

The resulting algorithm is the approximated Gauss-Newton algorithm, with update equation :

$$\underline{\boldsymbol{g}}_k(m+1) = \underline{\boldsymbol{g}}_k(m) - \frac{\mu}{2}\boldsymbol{\mathcal{W}}\underline{\boldsymbol{\nabla}}_{\underline{\boldsymbol{g}}_k} J_k\left(\underline{\boldsymbol{g}}_k\right) \quad (13)$$

The corresponding optimal step version of this algorithm is still obtained by using equation 11 with :

$$\underline{\boldsymbol{c}}_k(m) = \boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{H}}^T\underline{\boldsymbol{e}}_k(m)$$

The successive steps of this algorithm are summarized in Table 2.

It can be noted that a better candidate than the null equalizer can be chosen for the equalizer initialization for the two algorithms :

$$\underline{\boldsymbol{g}}_{k_0} = \mathcal{M}^T\mathcal{F}_J^{-1}\left(\mathbf{H}^H\mathbf{H} + \lambda\mathrm{Id}\right)^{-1}\mathbf{H}^H\underline{\mathbf{a}}_k \quad (14)$$

This initial equalizer is the one used in traditionnal FFT-based methods using regularization [3].

<table>
<tr><th colspan="2">Initialization</th></tr>
<tr><td colspan="2">Frequency Direct Matrix
$(\mathbf{H})_{i,j} = \mathrm{diag}\left(\mathcal{F}\mathcal{M}_h \underline{\boldsymbol{h}}_{ij}\right)$
Reference Vector
$\underline{\mathbf{a}}_k = \mathcal{F}_I \underline{\boldsymbol{a}}_k$
Frequency Inverse Autocorrelation Matrix
$\mathbf{W} = \left(\mathbf{H}^H\mathbf{H} + \lambda\mathrm{Id}\right)^{-1}$</td></tr>
<tr><th colspan="2">Iteration</th></tr>
<tr><td colspan="2">Estimated vector
$\underline{\hat{\mathbf{a}}}_k(m) = \mathbf{H}\mathcal{F}_J\mathcal{M}\underline{\boldsymbol{g}}_k(m)$
Error vector
$\underline{\mathbf{e}}_k(m) = \underline{\mathbf{a}}_k(m) - \underline{\hat{\mathbf{a}}}_k(m)$
Gradient vector
$\underline{\boldsymbol{d}}_k(m) = \mathcal{M}^T\mathcal{F}_J^{-1}\mathbf{H}^H\underline{\mathbf{e}}_k(m)$
Modified Gradient vector
$\underline{\boldsymbol{b}}_k(m) = \mathcal{M}^T\mathcal{F}_J^{-1}\mathbf{W}\mathcal{F}_J\mathcal{M}\underline{\boldsymbol{d}}_k(m)$</td></tr>
<tr><td>Normal Version</td><td>Optimal Step Version</td></tr>
<tr><td>Update
$\underline{\boldsymbol{g}}_k(m+1) = \underline{\boldsymbol{g}}_k(m) + \mu\underline{\boldsymbol{b}}_k(m)$</td><td>Optimal Step Vector
$\underline{\mathbf{c}}_k(m) = \mathbf{H}\mathcal{F}_J\mathcal{M}\underline{\boldsymbol{b}}_k(m)$
Iteration Step
$\mu(m) = \dfrac{\underline{\mathbf{e}}_k^H(m)\underline{\mathbf{c}}_k(m)}{\underline{\mathbf{c}}_k^H(m)\underline{\mathbf{c}}_k(m)}$
Update
$\underline{\boldsymbol{g}}_k(m+1) = \underline{\boldsymbol{g}}_k(m) + \mu(m)\underline{\boldsymbol{b}}_k(m)$</td></tr>
</table>

Table 2. Approximated Gauss-Newton Algorithms

4. RESULTS

In this section, results are presented in the case of typical transaural systems ($I = J = K = 2$). The equalization problem is known as cross-talk cancellation. The equalizer aims to compensate for both the room and the the loudspeakers responses, and also to cancel the cross-talk from the left loudspeaker to the right ear and vice-versa.

The direct transfert matrix H has been measured by sweep techniques [8]. The corresponding responses are represented on figure 2. The responses have been truncated to $L_h = 8192$ samples. The length of each inverse filter has been fixed to $L_g = 8193$ coefficients so that the convolution of the equalizer and the direct transfer matrix is of length 16384 samples.

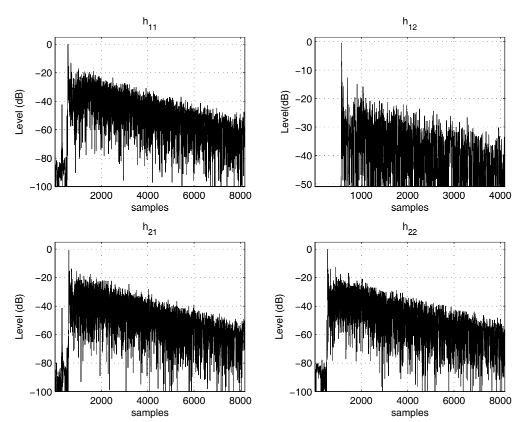

Fig. 2. Direct Transfert Matrix

Results of traditionnal FFT-based methods, synthesized by the equalizer of equation 14 are depicted on figure 3. The best results have been obtained with $\lambda = 10^{-2}$ for the regularization parameter. Nevertheless pre-echos and post-echos are visible (presence of secondary peaks before and after the main peak, approximately -35dB level). These artifacts are present because regularization isn't sufficient to limit the temporal extent of the inverse filters and then cannot cope with the time aliasing due to circular convolutions.

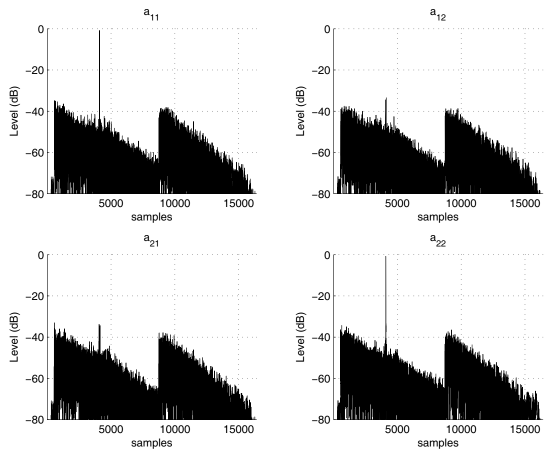

Fig. 3. Results with traditionnal equalization methods

Better performance could be achieved by using more coefficients in the inverse filters, or by using the iterative algorithms presented in this paper. Time aliasing doesn't occur in the inverse filter because linear convolutions are properly computed in the frequency domain, using FFT algorithms with correctly zero-padded sequences. The corrected system is presented on figure 4, after the convergence of the algorithm. Instead of secondary peaks, a quasi-constant noise reconstruction, approximatively -50 dB level, is visible.

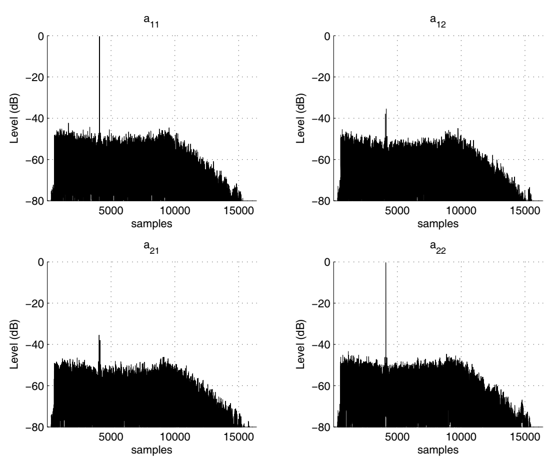

Fig. 4. Results after convergence of the algorithm

Concerning the performance of these algorithms, the MSE is represented on figure 5 for the optimal step gradient algorithm and for several values of λ for the optimal step approximated Gauss-Newton algorithm. It is seen that the accuracy of the obtained solution is better when the regularization parameter is low. On the other hand, the initial convergence speed is better when the regularization parameter is high. $\lambda = 10^{-2}$ seems to be a good tradeoff between speed convergence and accuracy of the solution. Note that this value corresponds to the one used in the traditionnal deconvolution method. Moreover, it is seen that approximated Gauss-Newton algorithm outperforms gradient algorithm concerning both speed convergence and accuracy of the solution.

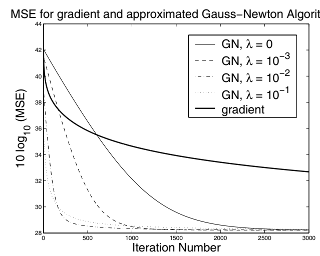

Fig. 5. Convergence properties of optimal step algorithms

5. CONCLUSION

Iterative multichannel equalization algorithms have been presented in this paper. They enable the inversion of long multiple electroacoustic impulse responses. Results have been given in the case of classical transaural applications and demonstrate the efficiency and accuracy of our approach, which is very suitable when optimal performance is required with a moderate number of coefficients in the inverse filters. It is important to emphasize that the proposed algorithms apply for any number of control points and/or loudspeakers.

6. REFERENCES

[1] M. Miyoshi and Y. Kaneda, "Inverse control of room acoustics using multiple loudspeakers and/or microphones," *ICASSP*, 1986.

[2] C. Breining and al., "Acoustic echo control : An application of very-high order adaptive filters," *IEEE Signal Processing Magazine*, 1999.

[3] O. Kirkeby and al., "Fast deconvolution of multichannel systems using regularization," *IEEE Transactions on Speech and Audio Processing*, 1998.

[4] O. Kirkeby and P.A. Nelson, "Digital filter design for inversion problems in sound reproduction," *Journal of Audio Engineering Society*, 1999.

[5] J. Bauck and D.H. Cooper, "Generalized transaural stereo and applications," *Journal of the Audio Engineering Society*, 1996.

[6] G.-O. Glentis and al., "Efficient least squares adaptive algorithms for fir transversal filtering," *IEEE Signal Processing Magazine*, 1999.

[7] J.J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Magazine*, 1992.

[8] S. Müller, "Transfer-function measurement with sweeps," *Journal of Audio Engineering Society*, 2001.