New Iterative Kernel Algorithms for Nonlinear Acoustic Echo Cancellation

Felix Albu[*], Kiyoshi Nishikawa[†]
[*]Department of Electronics, Valahia University of Targoviste, Targoviste, Romania
E-mail: felix.albu@valahia.ro; Tel: +40-245-217-683
[†]Department of System Design, Tokyo Metropolitan University, Tokyo, Hino-shi 191-0065, Japan
E-mail: knishikawa@m.ieice.org; Tel:

Abstract—**Recently, a nonlinear acoustic echo cancellation algorithm based on the framework of kernel methods has been proposed by modeling the echo path as a Hammerstein system. However, it requires a large amount of computation to be implemented. In this paper, we propose to use iterative methods for solving linear systems in order to reduce the numerical complexity of identifying the nonlinear and linear parts of the echo path. Also we investigate the effect on performance of the parameters of the methods for both iterative batch and online versions. Simulation results confirm the complexity reduction and good performance of the proposed methods for clipping nonlinearity and room impulse response variation.**

I. Introduction

The acoustic echo cancelling (AEC) systems are very useful in removing the acoustic echoes and allowing teleconferencing and videoconferencing [1]. The echo path depends on different factors such as position of the microphone, loudspeaker, objects in the room and temperature [1]. AECs include an adaptive filter in order to adaptively cancel the echo. In the absence of nonlinear distortions the adaptive filters using the affine projection algorithms can provide a good solution (e.g. [2]-[5]). However, in many cases low-cost loudspeakers are used and they cause nonlinear distortions. In the presence of nonlinear distortions the linear AECs are not sufficient to remove the nonlinear distortions [1], [6]–[8]. It is known that the most important nonlinear distortions are caused by the action of the loudspeaker and the power amplifier.

Many techniques have been proposed in the literature on nonlinear acoustic echo cancelation (NAEC). One of the most mentioned approaches is based on computationally expensive adaptive Volterra filters (VF) [8] or simplified VF [9]. Other proposed structures include even mirror Fourier nonlinear filters [10], block-based Wiener-Hammerstein models [11], functional link adaptive filters [12], etc. Other alternatives were mentioned in [13].

The kernel methods [14]-[15] have been proposed recently for NAEC. They still have a high computational complexity, although it is possible to limit the filter grow for *online* kernel methods [15]. In [13] the modeling of the echo path as a Hammerstein system and kernel-based identification algorithm for this model were proposed. It was shown that the input-output relationship was learnt with fewer data and less computational resources [13]. The method from [13] involves many inversions of very large matrices, therefore the numerical complexity is high. We propose to use iterative methods in order to reduce the computational load associated with matrices inversions. This idea is inspired from the resemblance between the efficient implementation of the kernel affine projection algorithm of [16]. The matrix inversions are replaced by linear system solving iterative methods. The use of Gauss-Seidel (GS) [17] and Conjugate Gradient (CG) [17] methods is proposed and the complexity/performance compromise is investigated. Therefore the complexity is reduced from $O\left(N^3\right)$ to $O\left(N^2\right)$ were N is the size of the matrix [18]. A novel idea of switching from GS to CG is also proposed.

The paper is organized as follows: Section II includes a short presentation of the Hammerstein system identification algorithm based on kernel methods from [13]. In section III the proposed methods are presented while in section IV the simulation results for a nonlinear acoustic echo cancellation scenario are shown. Section V presents the conclusions of this work and future work.

II. The Proposed Iterative Methods

Kernel methods are based on a nonlinear transformation of the input data x, into a high-dimensional *feature space*. The inner products in this space can be calculated by using a positive definite kernel function satisfying Mercer's condition, so called the "kernel trick" $k\left(x,x'\right)=\left\langle \Phi\left(x\right),\Phi\left(x'\right)\right\rangle$ [19]. A commonly used kernel function is the Gaussian kernel $k\left(x,x'\right)=\exp\left(-\left|x-x'\right|^2/2\sigma^2\right)$, where σ is the kernel width. In [13] a subset of the training data called support points, x_m^s, were used to represent the projection $\mathbf{w}=\sum_{m=1}^{M}\alpha_m\Phi\left(x_m^s\right)$.

The output of the estimated nonlinear function for a test input x' is the following [13]

$$f(x') = \Phi(x')^T \mathbf{w} = \sum_{m=1}^{M} k(x', x_m^s)\alpha_m \quad (1)$$

The $\boldsymbol{\alpha}$ coefficients are given by [13]

$$\boldsymbol{\alpha} = \left(\mathbf{K}^T\mathbf{K} + c\mathbf{K}_s\right)^{-1}\mathbf{K}^T\mathbf{d} \quad (2)$$

where $\mathbf{K}$ is the kernel matrix with elements $\mathbf{K}_{nm} = k(x(n), x_m^s)$, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_M]^T$ and $\mathbf{K}_s$ is the $M \times M$ kernel matrix calculated for support points [19].
In the following lines we have $\mathbf{d} = [d(1),...,d(N)]^T$, $\mathbf{h} = [h(1),...,h(L)]^T$ and the operator * refers to the convolution of $\mathbf{h}$ on the columns of $\mathbf{K}$, $\mathbf{K}_h = \hat{\mathbf{h}} * \mathbf{K}$ [13].

A. Batch identification algorithm

The observed echo path is modeled as a Hammerstein system, i.e., a cascade of nonlinearity and a linear filter [13]. The output of the microphone is given by

$$d(n) = h(n) * c(x(n)) + e_o(n) \quad (3)$$

where $e_o(n)$ is the background noise, $h(n)$ is the linear impulse response, including the room impulse response (RIR) of the echo path $h_{RIR}(n)$ and other linear parts, and $c(\cdot)$ represents the nonlinear distortion.

The cost function that is minimized is [13]

$$J = \|\mathbf{d} - \mathbf{h} * \mathbf{K}\boldsymbol{\alpha}\|^2 + c_\alpha \boldsymbol{\alpha}^T\mathbf{K}_s\boldsymbol{\alpha} + c_h\mathbf{h}^T\mathbf{h}. \quad (4)$$

For this case, the optimum solution for the non-linear filter coefficients $\hat{\boldsymbol{\alpha}}$ is given by [13]

$$\hat{\boldsymbol{\alpha}} = \left(\mathbf{K}_h^T\mathbf{K}_h + c_\alpha\mathbf{K}_s\right)^{-1}\mathbf{K}_h^T\mathbf{d}. \quad (5)$$

On the other hand, also shown in [13], that the solution for the linear filter coefficients $\hat{\mathbf{h}}$ is

$$\hat{\mathbf{h}} = \left(\mathbf{K}_\alpha^T\mathbf{K}_\alpha + c_h\mathbf{I}\right)^{-1}\mathbf{K}_\alpha^T\mathbf{d} \quad (6)$$

where $\mathbf{K}_\alpha$ contains the elements of $\mathbf{k}_\alpha = \mathbf{K}\hat{\boldsymbol{\alpha}}$ as follows [13]

$$\mathbf{K}_\alpha = \begin{bmatrix} k_\alpha(1) & 0 & ... & 0 \\ k_\alpha(2) & k_\alpha(1) & ... & 0 \\ \vdots & \vdots & & \vdots \\ k_\alpha(N) & k_\alpha(N-1) & ... & k_\alpha(N-L+1) \end{bmatrix} \quad (7)$$

The kernel-based identification of Hammerstein systems (KIHAM) algorithm for the batch identification alternates between updating estimates of the linear filter coefficients $\mathbf{h}$ and the nonlinear coefficients $\boldsymbol{\alpha}$ [13]. The initialization is made by setting the coefficients $\mathbf{h}$ to

$$\hat{\mathbf{h}} = \left(\mathbf{X}^T\mathbf{X} + c_h\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{d} \quad (8)$$

The KIHAM algorithm is shown in Table 1 [13]:

Table 1. The KIHAM Algorithm

Initialize $\mathbf{h}$ using Eq. 8
While *J* not converged **do**
Update $\mathbf{K}_h$ and $\hat{\boldsymbol{\alpha}}$ using Eq. 5
Update $\mathbf{K}_\alpha$ and $\hat{\mathbf{h}}$ using Eq. 7 and Eq. 6
end while
Output: $\hat{\boldsymbol{\alpha}}$ and $\hat{\mathbf{h}}$

The number of updates of the KIHAM algorithm depends on the *J* error threshold. In applications, the end condition of the while loop is given as $J < T_h$ where T_h is a threshold value [13]. Also, the number of updates can be set to maximum number depending on afforded computations in the **while**-loop.

B. Online strategy

An extension of the batch algorithm for application in online and adaptive scenarios was also proposed in [13]. It is considered that once the nonlinearity is estimated using the batch algorithm only the linear path is tracked. The online strategy is the following [13]:

1. A linear filter is used to obtain the linear coefficients $\hat{\mathbf{h}}(n)$.
2. Simultaneously a buffer is filled with $x(n)$ and $d(n)$.
3. The batch algorithm is used to obtain $\hat{\boldsymbol{\alpha}}$ when the buffer is filled with N_b input and output data, and $\hat{\mathbf{h}}(n)$ is updated.
4. The linear filter is updated using $f(x(n))$ and output $d(n)$.

III. The Proposed Algorithms

The KIHAM method involves the inversion of big matrices at each iteration ($\left(\mathbf{K}_h^T\mathbf{K}_h + c_\alpha\mathbf{K}_s\right)$ has a $M \times M$ size, where *M* is the size of the dictionary, while $\left(\mathbf{X}^T\mathbf{X} + c_h\mathbf{I}\right)$ has a $L \times L$ size, where *L* is the length of the filter). It is known that the inversion of a matrix requires $O(N^3)$ operations where *N* is the size of the matrix. Instead, in this paper, we consider using iterative algorithms in order to alleviate the known numerical instability of inverting large matrices [13].

Let us rewrite Eqs (5) and (6) of the KIHAM method in another form. If we note by $\mathbf{P}_h = \mathbf{K}_h^T\mathbf{d}$ and

$\mathbf{A}_h = \mathbf{K}_h^T \mathbf{K}_h + c_\alpha \mathbf{K}_s$ in Eq (5), we have $\hat{\boldsymbol{\alpha}} = \mathbf{A}_h^{-1}\mathbf{P}_h$ or alternatively $\hat{\boldsymbol{\alpha}}$ is the solution of the following system: $\mathbf{A}_h\hat{\boldsymbol{\alpha}} = \mathbf{P}_h$. Also, in Eq (6), if we note by $\mathbf{P}_\alpha = \mathbf{K}_\alpha^T\mathbf{d}$ and $\mathbf{A}_\alpha = \mathbf{K}_\alpha^T\mathbf{K}_\alpha + c_h\mathbf{I}$ we have $\hat{\mathbf{h}} = \mathbf{A}_\alpha^{-1}\mathbf{P}_\alpha$ or alternatively $\hat{\mathbf{h}}$ is the solution of the following system: $\mathbf{A}_\alpha\hat{\mathbf{h}} = \mathbf{P}_\alpha$.

We propose to use iterative methods to solve the linear systems $\mathbf{A}_h\hat{\boldsymbol{\alpha}} = \mathbf{P}_h$ and $\mathbf{A}_\alpha\hat{\mathbf{h}} = \mathbf{P}_\alpha$ and investigate the advantages and disadvantages regarding the convergence speed and numerical complexity.

The first iterative method is the Gauss-Seidel (GS) method. If we have to solve $\mathbf{A}x = \mathbf{b}$, where $\mathbf{A}$ is an $N \times N$ symmetric matrix, the Gauss-Seidel method computes at *k*-th iteration

$$x_i^{(k)} = \left(b_i - \sum_{j<i} a_{ij} x_j^{(k)} - \sum_{j>i} a_{ij} x_j^{(k-1)} \right) / a_{ii}. \qquad (9)$$

It is known that if the matrix A is symmetric and positive definite, the GS iteration is guaranteed to converge [17]. The GS algorithm has $N_i\ N^2$ multiplications, where N_i is the number of GS iterations.

The second method is the conjugate gradient (CG) algorithm [17] which is an example of exact line search method (In Table 2, we summarized the algorithm). The CG method has been used in adaptive filtering because of its fast convergence to the solution of normal equations. Its convergence is guaranteed to converge in no more than N steps. The number of multiplications of CG algorithm is $N_c\ (N^2+5N)$ multiplications where N_c shows the number of CG iterations [20]. Note that the numerical complexity of the CG algorithm is higher than that of the GS algorithm if they converge at the same number of iterations, i.e., $N_i = N_c$.

Table 2. The conjugate gradient (CG) algorithm

Step	
1	Initialization: $\hat{\mathbf{x}} = \mathbf{0}, \mathbf{r} = \mathbf{b}, \rho_0 = \mathbf{r}^T\mathbf{r}, \mathbf{d} = \mathbf{r}$
2	for k=1, ... , N_c
3	if k>1, $\mathbf{d} = \mathbf{r} + \frac{\rho_{k-1}}{\rho_{k-2}}\mathbf{d}$
4	$\mathbf{v}=\mathbf{A}\mathbf{d}$
5	$\gamma = \rho_{k-1} / \left(\mathbf{d}^T\mathbf{v}\right)$
6	$\hat{\mathbf{x}} = \hat{\mathbf{x}} + \gamma\mathbf{d}$
7	$\mathbf{r} = \mathbf{r} - \gamma\mathbf{v}$
8	$\rho_k = \mathbf{r}^T\mathbf{r}$

Like in [21] the solution for the previous iteration is used as an initial approximation of the solution for the actual iteration for both CG and GS methods. This is an essential step in assuring the use of a reduced number of iterations [21]. In this paper, we call the newly proposed algorithms as KIHAM-GS if using GS, and KIHAM-CG if using CG, respectively. In the following section we examine the influence of the number of iterations for both iterative methods. We also propose a novel idea to alternate the GS and CG methods in order to speed up the convergence towards the exact solution. For this case a new algorithm called KIHAM-GS-CG is proposed. For this algorithm the solution to each linear system of KIHAM is found as follows: firstly N_i GS iterations are performed and then N_c CG iterations are made. The intermediate solution after GS iterations is used as an initial approximation for the CG method. Its numerical complexity depends on the individual number of iterations for each iterative method. The computational complexity of KIHAM-GS-CG is between that of KIHAM-GS and KIHAM-CG for the same total number of GS and CG iterations. It can be considered that KIHAM-GS-CG reduces to KIHAM-GS if N_c=0, and to KIHAM-CG if N_i=0. The GS and CG methods were used among the numerous iterative techniques due to their good convergence properties and low numerical complexity [17]-[18]. At each time iteration, if sufficient iterations are made, the iterative solution can lead in some cases to a lower error than the exact one, since the linear systems to be solved are different.

IV. Simulation Results and Consideration

The performance of the proposed method is examined for a nonlinear echo cancellation scenario. It is considered that the main source of nonlinearity is due to the clipping of the power amplifier.

The signal $x(n)$ is generated as USASI noise with a speech-like spectrum [22]. The room impulse response (RIR) that describes the acoustic propagation $h_{RIR}(n)$ used in [13] were used. They had L = 512 taps and the sampling frequency was 8 kHz. The background noise was a zero-mean white Gaussian noise, and the signal to noise ratio at microphone location was 30 dB. As the performance index, we use the echo return loss enhancement (ERLE), expressed in decibels and defined as

$$\mathrm{ERLE}(n) = 10\log_{10}\frac{E\left\{d^2(n)\right\}}{E\left\{e^2(n)\right\}}. \qquad (10)$$

We evaluate the performance of the proposed algorithms considering an offline and an online scenarios.

C. *Offline scenario*

In this example N = 2048 data pairs were used. The KIHAM algorithms used a Gaussian kernel with a kernel width of 0.25, $c_\alpha = 0.01$, $c_h = 1$ and M = 50 support points equally spaced over the range of the input data [13]. The stopping error was $T_h = 10^{-6}$ and the maximum update number was 25. We considered the case of KIHAM-GS (N_i=3), KIHAM-CG (N_c=3) and KIHAM-GS-CG (one GS iteration and 2 CG iterations). As expected, the performance of KIHAM using direct methods is better than those of the proposed iterative KIHAM versions due to the used low number of GS or CG iterations. Similar results are obtained when using $N = 768$.

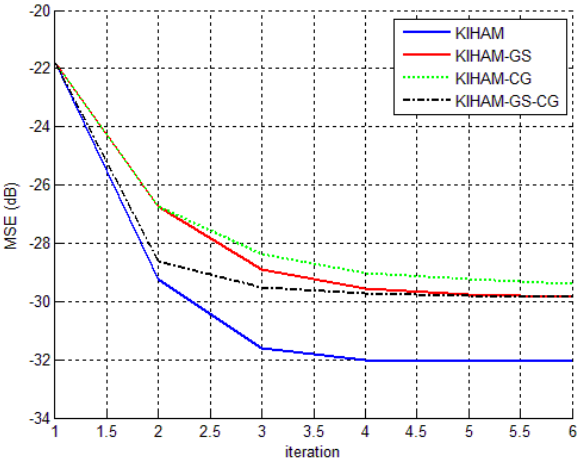

Fig. 1. MSE performance in case of an offline cancellation in case of 6 iterations for KIHAM, KIHAM-GS (N_i=3), KIHAM-CG (N_c =3) and KIHAM-GS-CG (N_i=1, N_c =2)

D. *Online scenario*

The ERLE performance of the proposed algorithms is investigated for an online scenario for 10 seconds, at 8 kHz. The tracking abilities are tested by an abrupt change of the RIR after 5 seconds. The parameters of the algorithms are the same except N = 768. The normalized least mean square (NLMS) algorithm with a unity step size is used for the linear part.

In the next figure, the influence of the number of iterations for KIHAM-CG is shown. It can be seen that KIHAM-CG with one iteration does not obtain good results in comparison with KIHAM.

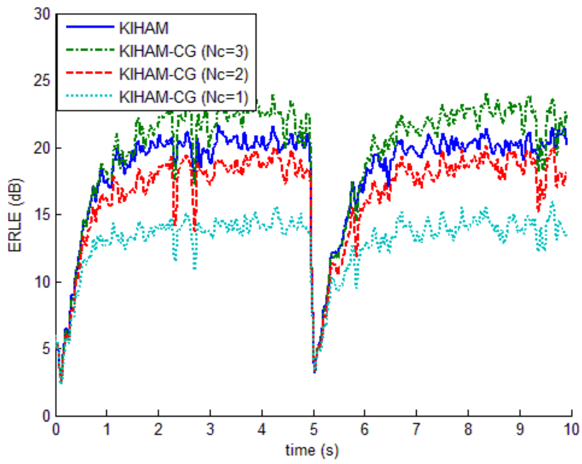

Fig. 2. Comparison of ERLE performance between KIHAM and KIHAM-CG for different number of CG iterations

It is obvious that better results are obtained if a higher number of iterations is performed, three CG iterations already obtaining better ERLE performance than KIHAM. The tracking abilities is not affected by the use of the iterative methods in the proposed algorithms.

In the next figure, the influence of the number of iterations for KIHAM-GS is shown using the same parameters as above. It can be seen that the increasing of the number of GS iterations does not have the same effect as the similar number of CG iterations.

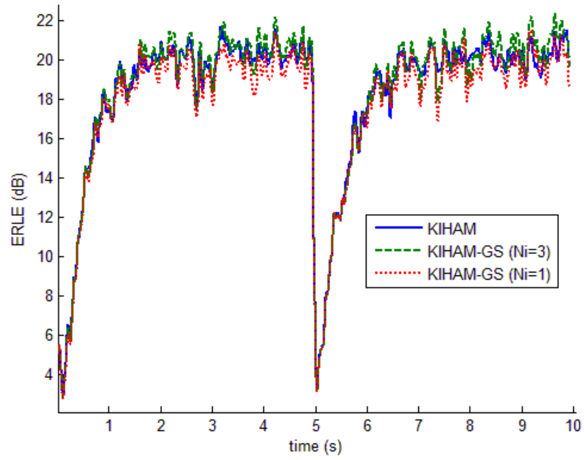

Fig. 3. Comparison of ERLE performance between KIHAM and KIHAM-GS for different number of GS iterations.

However, the performance of KIHAM-GS with one GS iteration is closer to that of KIHAM if compared with KIHAM-CG with one CG iteration. At convergence the average ERLE performance of KIHAM-GS with three GS iterations is slightly higher than that of KIHAM. By comparing the ERLE performance from the last two figures, the idea of combining GS and CG iterations in order to reduce the computational complexity was also investigated. It can be seen from Figs. 2 and 3 that KIHAM-GS with one GS iteration obtains a better performance than KIHAM-CG with one CG iteration, but KIHAM-CG with three CG iterations has a higher ERLE performance than KIHAM-GS with a similar number of GS iterations.

E. *Performance of KIHAM-GS-CG*

Based on the consideration in the previous sub-section, we investigate the performance of the new algorithm KIHAM-GS-CG algorithm. In the following two figures, the ERLE performance difference of KIHAM and KIHAM-GS-CG (1 GS and 2 CG iterations) is compared with the ERLE performance difference of KIHAM and KIHAM-CG with N_c=3 CG iterations for 25 and 15 updates respectively. The positive values over the red line indicates better performance of the iterative KIHAM based algorithms. It can be seen from Figs. 4 and 5 that the reduction of complexity given by replacing one CG iteration with one GS iteration leads to a slight reduction of the ERLE performance. Also, it can be noticed when the number of updates for the linear and nonlinear coefficients decreases the performance of the proposed iterative KIHAM based algorithms also decreases.

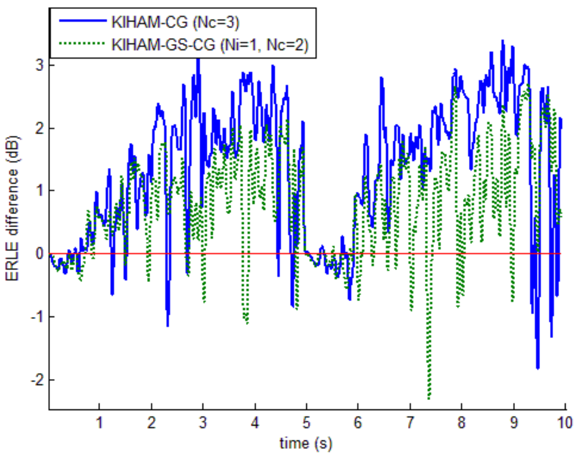

Fig. 4. Comparison of ERLE performance difference after 25 updates between KIHAM, KIHAM-CG ($N_c = 3$) and KIHAM-GS-CG ($N_i = 1$ and N_c=2) respectively (50 support points)

A similar behavior is observed when the number of support points x_m^s, $m = 1..M$ is reduced from 50 to 25 (see Fig. 6). Therefore, the performance of the proposed KIHAM based algorithms depends on the number of GS or CG iterations, N, and the number of support points. Our simulations have shown that the number of needed iterations for the proposed algorithms has to be increased if the N parameter increases.

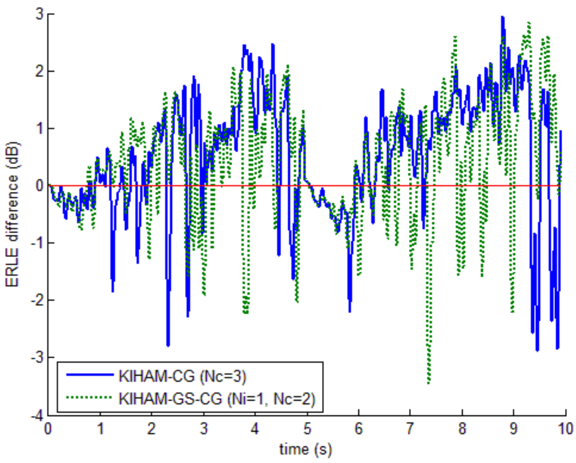

Fig. 5. Comparison of ERLE performance difference after 15 updates between KIHAM, KIHAM-CG ($N_c = 3$) and KIHAM-GS-CG ($N_i = 1$ and N_c=2) respectively (50 support points)

If only the saving due to the use of the iterative methods is taken into account, the savings in terms of multiplications is around 84% for the proposed KIHAM based algorithms (the parameters were $L = 512$, $M = 50$, N_c=3, 15 updates) and reduces to 74% if L = 128, while the other parameters are the same. The overall computational savings is smaller and depends on the algorithm parameters and efficiency of the practical implementation.

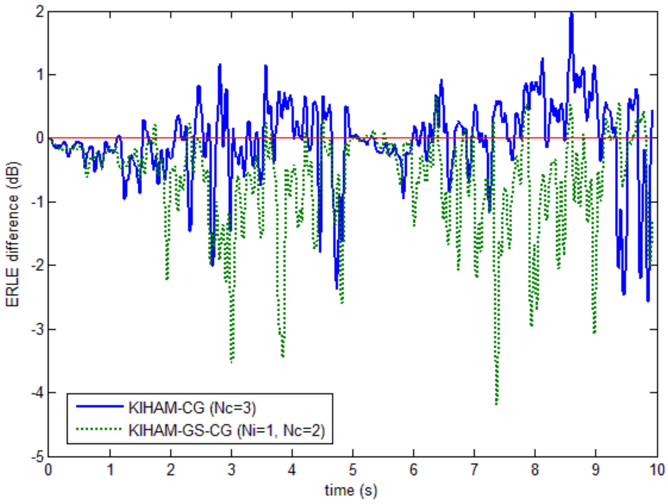

Fig. 6. Comparison of ERLE performance difference after 25 updates between KIHAM, KIHAM-CG ($N_c = 3$) and KIHAM-GS-CG ($N_i = 1$ and N_c=2) respectively (25 support points)

The performance of the proposed algorithms depends on the N parameter. If the value of N is increased to 2048 while keeping the same parameters from Fig. 6 the proposed algorithms are only having a close smaller performance (less than 0.5 dB difference) to the original KIHAM algorithm (see Fig. 7). Therefore, it can be concluded from Figs. 1-7 that the proposed iterative algorithms performance largely depends on the chosen parameters for each scenario.

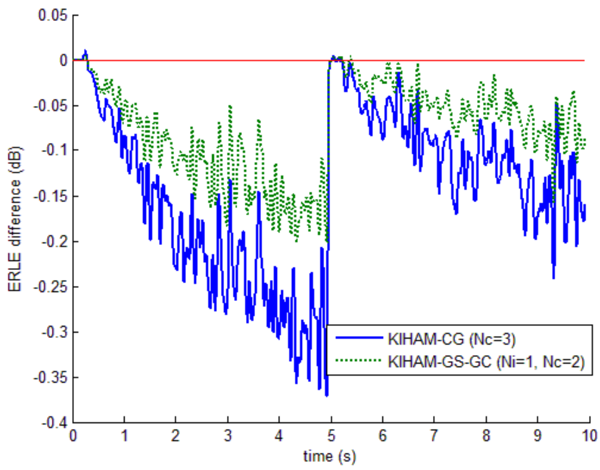

Fig. 7. Comparison of ERLE performance difference after 25 updates for N = 2048 between KIHAM, KIHAM-CG ($N_c = 3$) and KIHAM-GS-CG (N_i = 1 and N_c=2) respectively (25 support points)

V. CONCLUSIONS

This paper has proposed new adaptive algorithms for nonlinear echo cancellation that use the Gauss-Seidel and conjugate gradient methods instead of the direct matrix solving methods. The model assumes the clipping as a main distortion of the audio power amplifier and behavior of the loudspeaker and the echo path as being a Hammerstein system. Experiments confirm a good compromise complexity/performance with the original implementations for both batch and online scenarios. Our future will be focused on evaluating *the* number of necessary iterations of the GS or CG schemes for various nonlinearities, number of support points and filter lengths.

ACKNOWLEDGMENT

This work was supported by a grant of the Romanian National Authority for Scientific Research, CNCS-UEFISCDI, project number PN-II-ID-PCE-2011-3-0097.

REFERENCES

[1] C. Breining, P. Dreiscitel, E. Hansler, A. Mader, B. Nitsch, H. Puder, T. Schertler, G. Schmidt, and J. Tilp, "Acoustic echo control. An application of very-high-order adaptive filters," *IEEE Signal Processing Magazine*, vol. 16, no. 4, pp. 42–69, 1999.

[2] F. Albu, C. Paleologu and J. Benesty, "A Variable Step Size Evolutionary Affine Projection Algorithm," in Proc. of ICASSP 2011, May 2011, pp. 429-432.

[3] F. Albu, C. Paleologu, J. Benesty, and S. Ciochina, "A low complexity proportionate affine projection algorithm for echo cancellation," in *Proc. of EUSIPCO 2010*, pp. 6–10.

[4] A. Gonzalez, M. Ferrer, F. Albu, and M. de Diego, "Affine projection algorithms: evolution to smart and fast multichannel algorithms and applications", in *Proc. of Eusipco 2012*, August 2012, pp. 1965-1969.

[5] F. Albu and H. K. Kwan, "Memory Improved Proportionate Affine Projection Sign Algorithm", *IET Electronics Letters*, vol. 48, no. 20, pp. 1279-1281, October 2012.

[6] M. Mossi, N. Evans, and C. Beaugeant, "An assessment of linear adaptive filter performance with nonlinear distortions," in *Proc. of Acoustics, Speech, and Signal Processing*, 2010, pp. 313–316.

[7] L. Azpicueta-Ruiz, M. Zeller, A. Figueiras-Vidal, J. Arenas-Garcıa, and W. Kellermann, "Adaptive combination of Volterra kernels and its application to nonlinear acoustic echo cancellation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 97–110, 2011.

[8] F. Kuch and W. Kellermann, *Nonlinear Acoustic Echo Cancellation*, Springer, Heidelberg, Germany, 2006.

[9] F. Albu, C. Kotropoulos, "Modified Gauss-Seidel affine projection algorithm for acoustic echo cancellation", in *Proc. of ICASSP 2005*, pp. 121-124, 2005.

[10] A. Carini and G. Sicuranza, "Fourier nonlinear filters," *Signal Processing*, vol. 94, pp. 183–194, 2014.

[11] J. Voros, "An iterative method for Hammerstein-Wiener systems parameter identification," *Journal of Electrical Engineering*, vol. 55, pp. 328–331, 2004.

[12] D. Comminiello, M. Scarpiniti, L. Azpicueta-Ruiz, J. Arenas-Garcıa, and A. Uncini, "Functional link adaptive filters for nonlinear acoustic echo cancellation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1502–1512, 2013.

[13] S. Van Vaerenbergh, and L. Azpicueta-Ruiz, "Kernel-based identification of Hammerstein systems for nonlinear acoustic echo-cancellation," in *Proc. of ICASSP 2014,* pp. 3739-3743, 2014.

[14] B. Scholkopf and A. J. Smola, *Learning with Kernels*, The MIT Press, Cambridge, MA, USA, 2002

[15] J. Gil-Cacho, M. Signoretto, T. van Waterschoot, M. Moonen, and S. Jensen, "Nonlinear acoustic echo cancellation based on a sliding-window leaky kernel affine projection algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 9, pp. 1867–1878, 2013.

[16] F. Albu, D. Coltuc, K. Nishikawa, M. Rotaru, "An efficient implementation of the kernel affine projection algorithm," in *Proc. of IEEE ISPA 2013*, September 2013, Trieste, Italy, pp. 342-346

[17] R.Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. van der Vorst, "*Templates for the solutions of linear systems: Building blocks for iterative methods*," SIAM, 1994.

[18] G. H. Golub and C. F. Van Loan, *Matrix computation*, 3rd edition. Baltimore, MD: The John Hopkins Univ. Press, 1996.

[19] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag New York, Inc., 1995.

[20] Y. Zakharov, G. White, and J. Liu, "Low complexity RLS algorithms using dichotomous coordinate descent iterations," *IEEE Trans. Signal Processing*, vol.56. No.7, pp. 3150-3161, 2008

[21] F. Albu, A. Fagan, "The Gauss-Seidel pseudo affine projection algorithm and its application for echo cancellation, Conference Record of *the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, 2003, Vol. 2, 9-12 Nov. 2003, pp. 1303 – 1306.

[22] "NRSC AM preemphasis/deemphasis and broadcast audio transmission bandwidth specifications," *Standard ANSI/EIA-54988*, 1998.