Adaptive Partial-Update and Sparse System Identification

Guest Editors: Kutluyıl Doğançay and Patrick A. Naylor
Adaptive Partial-Update and Sparse System Identification
Adaptive Partial-Update and Sparse System Identification

Guest Editors: Kutluyl Doğançay and Patrick A. Naylor
Editor-in-Chief
Douglas O’Shaughnessy, University of Quebec, Canada

Associate Editors
Jont B. Allen, USA
Xavier Amatriain, USA
Gérard Bailly, France
Martin Bouchard, Canada
Douglas S. Brungart, USA
Geoffrey Chan, Canada
Dan Chazan, Israel
Mark Clements, USA
C. D’alessandro, France
Roger Dannenberg, USA
Li Deng, USA
Thomas Eriksson, Sweden

Horacio Franco, USA
Qian-Jie Fu, USA
Jim Glass, USA
Steven Greenberg, USA
R. Capobianco Guido, Brazil
R. Heusdens, The Netherlands
James Kates, USA
Tatsuya Kawahara, Japan
Yves Laprie, France
Lin-Shan Lee, Taiwan
Dominic Massaro, USA
Ben Milner, USA

Climent Nadeu, Spain
Elmar Noth, Germany
Hiroshi Okuno, Japan
Joe Picone, USA
Gerhard Rigoll, Germany
Mark Sandler, UK
Thippur V. Sreenivas, India
Yannis Stylianou, Greece
Stephen Voran, USA
Deliang Wang, USA
Contents

Adaptive Partial-Update and Sparse System Identification, Kutluyıl Doğançay and Patrick A. Naylor
Volume 2007, Article ID 12046, 2 pages

Set-Membership Proportionate Affine Projection Algorithms, Stefan Werner, José A. Apolinário, Jr., and Paolo S. R. Diniz
Volume 2007, Article ID 34242, 10 pages

Wavelet-Based MPNLMS Adaptive Algorithm for Network Echo Cancellation, Hongyang Deng and Miloš Doroslovački
Volume 2007, Article ID 96101, 5 pages

A Low Delay and Fast Converging Improved Proportionate Algorithm for Sparse System Identification, Andy W. H. Khong, Patrick A. Naylor, and Jacob Benesty
Volume 2007, Article ID 84376, 8 pages

Analysis of Transient and Steady-State Behavior of a Multichannel Filtered-x Partial-Error Affine Projection Algorithm, Alberto Carini and Giovanni L. Sicuranza
Volume 2007, Article ID 31314, 15 pages

Step Size Bound of the Sequential Partial Update LMS Algorithm with Periodic Input Signals, Pedro Ramos, Roberto Torrubia, Ana López, Ana Salinas, and Enrique Masgrau
Volume 2007, Article ID 10231, 15 pages

Detection-Guided Fast Affine Projection Channel Estimator for Speech Applications, Yan Wu Jennifer, John Homer, Geert Rombouts, and Marc Moonen
Volume 2007, Article ID 71495, 13 pages

Efficient Multichannel NLMS Implementation for Acoustic Echo Cancellation, Fredric Lindstrom, Christian Schüldt, and Ingvar Claesson
Volume 2007, Article ID 78439, 6 pages

Time-Domain Convolutive Blind Source Separation Employing Selective-Tap Adaptive Algorithms, Qiongfeng Pan and Tyseer Aboulnasr
Volume 2007, Article ID 92528, 11 pages

Underdetermined Blind Audio Source Separation Using Modal Decomposition, Abdeldjalil Aïssa-El-Bey, Karim Abed-Meraim, and Yves Grenier
Volume 2007, Article ID 85438, 15 pages
System identification is an important task in many application areas including, for example, telecommunications, control engineering, sensing, and acoustics. It would be widely accepted that the science for identification of stationary and dynamic systems is mature. However, several new applications have recently become of heightened interest for which system identification needs to be performed on high-order moving average systems that are either sparse in the time domain or need to be estimated using sparse computation due to complexity constraints. In this special issue, we have brought together a collection of articles on recent work in this field giving specific consideration to (a) algorithms for identification of sparse systems and (b) algorithms that exploit sparseness in the coefficient update domain. The distinction between these two types of sparseness is important, as we hope will become clear to the reader in the main body of the special issue.

A driving force behind the development of algorithms for sparse system identification in telecommunications has been echo cancellation in packet switched telephone networks. The increasing popularity of packet-switched telephony has led to a need for the integration of older analog systems with, for example, IP or ATM networks. Network gateways enable the interconnection of such networks and provide echo cancellation. In such systems, the hybrid echo response is delayed by an unknown bulk delay due to propagation through the network. The overall effect is, therefore, that an “active” region associated with the true hybrid echo response occurs with an unknown delay within an overall response window that has to be sufficiently long to accommodate the worst case bulk delay. In the context of network echo cancellation the direct application of well-known algorithms, such as normalized least-mean-square (NLMS), to sparse system identification gives unsatisfactory performance when the echo response is sparse. This is because the adaptive algorithm has to operate on a long filter and the coefficient noise for near-zero-valued coefficients in the inactive regions is relatively large. To address this problem, the concept of proportionate updating was introduced.

An important consideration for adaptive filters is the computational complexity that increases with the number of coefficients to be updated per sampling period. A straightforward approach to complexity reduction is to update only a small number of filter coefficients at every iteration. This approach is termed partial-update adaptive filtering. Two key questions arise in the context of partial updating. Firstly, consideration must be given as to how to choose which coefficients to update. Secondly, the performance and complexity of the partial update approach must be compared with the standard full update algorithms in order to assess the cost-to-benefit ratio for the partial update schemes. Usually, a compromise has to be made between affordable complexity and desired convergence speed.

We have grouped the papers in this special issue into four areas. The first area is sparse system identification and comprises three papers. In “Set-membership proportionate affine projection algorithms,” Stefan Werner et al. develop affine projection algorithms with proportionate update and set membership filtering. Proportionate updates facilitate fast convergence for sparse systems, and set membership filtering reduces the update complexity. The second paper in this area is “Wavelet-based MPNLMS adaptive algorithm for network echo cancellation” by H. Deng and M. Doroslovački, which develops a wavelet-domain $\mu$-law proportionate NLMS algorithm for identification and cancelling of sparse telephone network echoes. This work exploits the whitening and good time-frequency localisation properties of the wavelet transform to speed up the convergence for coloured input signals and to retain sparseness of echo response in the wavelet transform domain. In “A low delay and
fast converging improved proportionate algorithm for sparse system identification," Andy W. H. Khong et al. propose a multidelay filter (MDF) implementation for improved proportionate NLMS for sparse system identification, inheriting the beneficial properties of both; namely, fast convergence and computational efficiency coupled with low bulk delay. As the authors show, the MDF implementation is nontrivial and requires time-domain coefficient updating.

The second area of papers is partial-update active noise control. In the first paper in this area "Analysis of transient and steady-state behavior of a multichannel filtered-x partial-error affine projection algorithm," A. Carini and S. L. Sicuranza apply partial-error complexity reduction to filtered-x affine projection algorithm for multichannel active noise control, and provide a comprehensive analysis of the transient and steady-state behaviour of the adaptive algorithm drawing on energy conservation. In "Step size bound of the sequential partial update LMS algorithm with periodic input signals" Pedro Ramos et al. show that for periodic input signals the sequential partial update LMS and filtered-x LMS algorithms can achieve the same convergence performance as their full-update counterparts by increasing the step-size appropriately. This essentially avoids any convergence penalty associated with sequential updating.

The third area focuses on general partial update algorithms. In the first paper in this area, "Detection guided fast affine projection channel estimator for speech applications," Yan Wu Jennifer et al. consider detection guided identification of active taps in a long acoustic echo channel in order to shorten the actual channel and integrate it into the fast affine projection algorithm to attain faster convergence. The proposed algorithm is well suited for highly correlated input signals such as speech signals. In "Efficient multichannel NLMS implementation for acoustic echo cancellation," Fredric Lindstrom et al. propose a multichannel acoustic echo cancellation algorithm based on normalized least-mean-square with partial updates favouring filters with largest misadjustment.

The final area is devoted to blind source separation. In "Time domain convolutive blind source separation employing selective-tap adaptive algorithms," Q. Pan and T. Aboul-nasr propose time-domain convolutive blind source separation algorithms employing M-max and exclusive maximum selective-tap techniques. The resulting algorithms have reduced complexity and improved convergence performance thanks to partial updating and reduced interchannel coherence. In the final paper "Underdetermined blind audio source separation using modal decomposition," Abdeljalil Aissa-El-Bey et al. present a novel blind source separation algorithm for audio signals using modal decomposition. In addition to instantaneous mixing, the authors consider convolutive mixing and exploit the sparseness of audio signals to identify the channel responses before applying modal decomposition.

In summary, we can say that sparseness in the context of adaptive filtering presents both challenges and opportunities. Standard adaptive algorithms suffer a degradation in performance when the system to be identified is sparse. This has created the need for new algorithms for sparse adaptive filtering—a challenge that has been well met to date for the particular applications addressed. When sparseness exists, or can be safely assumed, in input signals, this can be exploited to achieve both computational savings in partial update schemes and, in certain specific cases, performance improvements. There remain several open research questions in this context and we look forward to an ongoing research effort in the scientific community and opportunities for algorithm deployment in real-time applications.

ACKNOWLEDGMENTS

This special issue has arisen as a result of the high levels of interest shown at a special session on this topic at EUSIPCO 2005 in Antalya, Turkey. It has been a great privilege to act as guest editors for this special issue and we extend our grateful thanks to all the authors and the publisher.

Kutluyıl Doğancay
Patrick A. Naylor
Set-Membership Proportionate Affine Projection Algorithms

Stefan Werner,1 José A. Apolinário, Jr.,2 and Paulo S. R. Diniz3

1 Signal Processing Laboratory, Helsinki University of Technology, Otakaari 5A, 02150 Espoo, Finland
2 Department of Electrical Engineering, Instituto Militar de Engenharia, 2229-270 Rio de Janeiro, Brazil
3 Signal Processing Laboratory, COPPE/Poli/Universidade Federal do Rio de Janeiro, 21945-970 Rio de Janeiro, Brazil

Received 30 June 2006; Revised 15 November 2006; Accepted 15 November 2006

Recommended by Kutluyil Dogancay

Proportionate adaptive filters can improve the convergence speed for the identification of sparse systems as compared to their conventional counterparts. In this paper, the idea of proportionate adaptation is combined with the framework of set-membership filtering (SMF) in an attempt to derive novel computationally efficient algorithms. The resulting algorithms attain an attractive faster converge for both situations of sparse and dispersive channels while decreasing the average computational complexity due to the data discerning feature of the SMF approach. In addition, we propose a rule that allows us to automatically adjust the number of past data pairs employed in the update. This leads to a set-membership proportionate affine projection algorithm (SM-PAPA) having a variable data-reuse factor allowing a significant reduction in the overall complexity when compared with a fixed data-reuse factor. Reduced-complexity implementations of the proposed algorithms are also considered that reduce the dimensions of the matrix inversions involved in the update. Simulations show good results in terms of reduced number of updates, speed of convergence, and final mean-squared error.

Copyright © 2007 Stefan Werner et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Frequently used adaptive filtering algorithms like the least mean square (LMS) and the normalized LMS (NLMS) algorithms share the features of low computational complexity and proven robustness. The LMS and the NLMS algorithms have in common that the adaptive filter is updated in the direction of the input vector without favoring any particular direction. In other words, they are well suited for dispersive-type systems where the energy is uniformly distributed among the coefficients in the impulse response. On the other hand, if the system to be identified is sparse, that is, the impulse response is characterized by a few dominant coefficients (see [1] for a definition of a measure of sparsity), using different step sizes for each adaptive filter coefficient can improve the initial convergence of the NLMS algorithm. This basic concept is explored in proportionate adaptive filters [2–10], which incorporates the importance of the individual components by assigning weights proportional to the magnitude of the coefficients.

The conventional proportionate NLMS (PNLMS) algorithm [2] experiences fast initial adaptation for the dominant coefficients followed by a slower second transient for the remaining coefficients. Therefore, the slow convergence of the PNLMS algorithm after the initial transient can be circumvented by switching to the NLMS algorithm [11].

Another problem related to the conventional PNLMS algorithm is the poor performance in dispersive or semi-dispersive channels [3]. Refinements of the PNLMS have been proposed [3, 4] to improve performance in a dispersive medium and to combat the slowdown after the initial adaptation. The PNLMS++ algorithm in [3] approaches the problem by alternating the NLMS update with a PNLMS update. The improved PNLMS (IPNLMS) algorithm [4] combines the NLMS and PNLMS algorithms into one single updating expression. The main idea of the IPNLMS algorithm was to establish a rule for automatically switching from one algorithm to the other. It was further shown in [6] that the IPNLMS algorithm is a good approximation of the exponentiated gradient algorithm [1, 12]. Extension of the proportionate adaptation concept to affine projection (AP) type algorithms, proportionate affine projection (PAP) algorithms, can be found in [13, 14].

Using the PNLMS algorithm instead of the NLMS algorithm leads to 50% increase in the computational complexity. An efficient approach to reduce computations is to employ set-membership filtering (SMF) techniques [15, 16], where the filter is designed such that the output estimation
error is upper bounded by a predetermined threshold. Set-membership adaptive filters (SMAF) feature data-selective (sparse in time) updating, and a time-varying data-dependent step size that provides fast convergence as well as low steady-state error. SMAFs with low computational complexity per update are the set-membership NLMS (SM-NLMS) [15], the set-membership binormalized data-reusing (SM-BNDRLMS) [16], and the set-membership affine projection (SM-AP) [17] algorithms. In the following, we combine the frameworks of proportionate adaptation and SMF. A set-membership proportionate NLMS (SM-PNLMS) algorithm is proposed as a viable alternative to the SM-NLMS algorithm [15] for operation in sparse scenarios. Following the ideas of the IPNLMS algorithm, an efficient weight-scaling assignment is proposed that utilizes the information provided by the data-dependent step size. Thereafter, we propose a more general algorithm, the set-membership proportionate affine projection algorithm (SM-PAPA) that generalizes the ideas of the SM-PNLMS to reuse constraint sets from a fixed number of past input and desired signal pairs in the same way as the SM-AP algorithm [17]. The resulting algorithm can be seen as a set-membership version of the PAP algorithm [13, 14] with an optimized step size. As with the PAP algorithm, a faster convergence of the SM-PAPA algorithm may come at the expense of a slight increase in the computational complexity per update that is directly linked to the amount of reuses employed, or data-reuse factor. To lower the overall complexity, we propose to use a time-varying data-reuse factor. The introduction of the variable data-reuse factor results in an algorithm that close to convergence takes the form of the simple SM-PNLMS algorithm. Thereafter, we consider an efficient implementation of the new SM-PAPA algorithm that reduces the dimensions of the matrices involved in the update.

The paper is organized as follows. Section 2 reviews the concept of SMF while the SM-PNLMS algorithm is proposed in Section 3. Section 4 derives the general SM-PAPA algorithm where both cases of fixed and time-varying data-reuse factor are studied. Section 5 provides the details of an SM-PAPA implementation using reduced matrix dimensions. In Section 6, the performances of the proposed algorithms are evaluated through simulations which are followed by conclusions.

2. SET-MEMBERSHIP FILTERING

This section reviews the basic concepts of set-membership filtering (SMF). For a more detailed introduction to the concept of SMF, the reader is referred to [18]. Set-membership filtering is a framework applicable to filtering problems that are linear in parameters. A specification on the filter parameters \( \mathbf{w} \in \mathbb{C}^N \) is achieved by constraining the magnitude of the output estimation error, \( e(k) = d(k) - \mathbf{w}^H \mathbf{x}(k) \), to be smaller than a deterministic threshold \( y \), where \( \mathbf{x}(k) \in \mathbb{C}^N \) and \( d(k) \in \mathbb{C} \) denote the input vector and the desired output signal, respectively. As a result of the bounded error constraint, there will exist a set of filters rather than a single estimate.

Let \( \delta \) denote the set of all possible input-desired data pairs \((\mathbf{x}, d)\) of interest. Let \( \Theta \) denote the set of all possible vectors \( \mathbf{w} \) that result in an output error bounded by \( y \) whenever \((\mathbf{x}, d) \in \delta \). The set \( \Theta \) referred to as the feasibility set is given by

\[
\Theta = \bigcap_{(\mathbf{x}, d) \in \delta} \{ \mathbf{w} \in \mathbb{C}^N : |d - \mathbf{w}^H \mathbf{x}| \leq y \}. \tag{1}
\]

Adaptive SMF algorithms seek solutions that belong to the exact membership set \( \psi(k) \) constructed by input-signal and desired-signal pairs,

\[
\psi(k) = \bigcap_{i=1}^{k} \mathcal{H}(i), \tag{2}
\]

where \( \mathcal{H}(k) \) is referred to as the constraint set containing all vectors \( \mathbf{w} \) for which the associated output error at time instant \( k \) is upper bounded in magnitude by \( y \):

\[
\mathcal{H}(k) = \{ \mathbf{w} \in \mathbb{C}^N : |d(k) - \mathbf{w}^H \mathbf{x}(k)| \leq y \}. \tag{3}
\]

It can be seen that the feasibility set \( \Theta \) is a subset of the exact membership set \( \psi_k \) at any given time instant. The feasibility set is also the limiting set of the exact membership set, that is, the two sets will be equal if the training signal traverses all signal pairs belonging to \( \delta \). The idea of set-membership adaptive filters (SMAF) is to find adaptively an estimate that belongs to the feasibility set or to one of its members. Since \( \psi(k) \) in (2) is not easily computed, one approach is to apply one of the many optimal bounding ellipsoid (OBE) algorithms [18, 20–24], which tries to approximate the exact membership set \( \psi(k) \) by tightly outer bounding it with ellipsoids. Adaptive approaches leading to algorithms with low peak complexity, \( O(N) \), compute a point estimate through projections using information provided by past constraint sets [15–17, 25–27]. In this paper, we are interested in algorithms derived from the latter approach.

3. THE SET-MEMBERSHIP PROPORTIONATE NLMS ALGORITHM

In this section, the idea of proportionate adaptation is applied to SMF in order to derive a data-selective algorithm, the set-membership proportionate normalized LMS (SM-PNLMS), suitable for sparse environments.

3.1. Algorithm derivation

The SM-PNLMS algorithm uses the information provided by the constraint set \( \mathcal{H}(k) \) and the coefficient updating to solve the optimization problem employing the criterion

\[
\mathbf{w}(k+1) = \arg \min_{\mathbf{w}} \| \mathbf{w} - \mathbf{w}(k) \|^2_{G^{-1}(k)} \quad \text{subject to:} \quad \mathbf{w} \in \mathcal{H}(k), \tag{4}
\]
where the norm employed is defined as $\|b\|_A = b^H A b$. Matrix $G(k)$ is here chosen as a diagonal weighting matrix of the form

$$G(k) = \text{diag}\{g_1(k), \ldots, g_N(k)\}.$$  

(5)

The elements values of $G(k)$ will be discussed in Section 3.2. The optimization criterion in (4) states that if the previous estimate already belongs to the constraint set, $w(k) \in \mathcal{H}(k)$, it is a feasible solution and no update is needed. However, if $w(k) \notin \mathcal{H}(k)$, an update is required. Following the principle of minimal disturbance, a feasible update is made such that $w(k + 1)$ lies up on the nearest boundary of $\mathcal{H}(k)$. In this case the updating equation is given by

$$w(k + 1) = w(k) + \alpha(k) \frac{e^+(k) G(k) x(k)}{x^H(k) G(k) x(k)},$$

(6)

where

$$\alpha(k) = \begin{cases} 
1 - \frac{\gamma}{|e(k)|} & \text{if } |e(k)| > \gamma \\
0 & \text{otherwise} \end{cases}$$

(7)

is a time-varying data-dependent step size, and $e(k)$ is the a priori error given by

$$e(k) = d(k) - w^H(k) x(k).$$

(8)

For the proportionate algorithms considered in this paper, matrix $G(k)$ will be diagonal. However, for other choices of $G(k)$, it is possible to identify from (6) different types of SMAF available in literature. For example, choosing $G(k) = I$ gives the SM-NLMS algorithm [15], setting $G(k)$ equal to a weighted covariance matrix will result in the BEACON recursions [28], and choosing $G(k)$ such that it extracts the $P \leq N$ elements in $x(k)$ of largest magnitude gives a partial-updating SMF [26]. Next we consider the weighting matrix used with the SM-PNLMS algorithm.

### 3.2. Choice of weighting matrix $G(k)$

This section proposes a weighting matrix $G(k)$ suitable for operation in sparse environments.

Following the same line of thought as in the IPNLMS algorithm, the diagonal elements of $G(k)$ are computed to provide a good balance between the SM-NLMS algorithm and a solution for sparse systems. The goal is to reduce the length of the initial transient for estimating the dominant peaks in the impulse response and, thereafter, to emphasize the input-signal direction to avoid a slow second transient. Furthermore, the solution should not be sensitive to the assumption of a sparse system. From the expression for $\alpha(k)$ in (7), we observe that, if the solution is far from the constraint set, we have $\alpha(k) \to 1$, whereas close to the steady state $\alpha(k) \to 0$. Therefore, a suitable weight assignment rule emphasizes dominant peaks when $\alpha(k) \to 1$ and the input-signal direction (SM-PNLMS update) when $\alpha(k) \to 0$. As $\alpha(k)$ is a good indicator of how close a steady-state solution is, we propose to use

$$g_i(k) = \frac{1 - \kappa \alpha(k)}{N} + \frac{\kappa \alpha(k) |w_i(k)|}{||w(k)||_1},$$

(9)

where $\kappa \in [0, 1]$ and $||w(k)||_1 = \sum |w_i(k)|$ denotes the $l_1$ norm [2, 4]. The constant $\kappa$ is included to increase the robustness for estimation errors in $w(k)$, and from the simulations provided in Section 6, $\kappa = 0.5$ shows good performance for both sparse and dispersive systems. For $\kappa = 1$, the algorithm will converge faster but will be more sensitive to the sparse assumption. The IPNLMS algorithm uses similar strategy, see [4] for details. The updating expressions in (9) and (6) resemble those of the IPNLMS algorithm except for the time-varying step size $\alpha(k)$. From (9) we can observe the following: (1) during initial adaptation (i.e., during transient) the solution is far from the steady-state solution, and consequently $\alpha(k)$ is large, and more weight will be placed at the stronger components of the adaptive filter impulse response; (2) as the error decreases, $\alpha(k)$ gets smaller, all the coefficients become equally important, and the algorithm behaves as the SM-NLMS algorithm.

### 4. THE SET-MEMBERSHIP PROPORTIONATE AFFINE-PROJECTION ALGORITHM

In this section, we extend the results from the previous section to derive an algorithm that utilizes the $L(k)$ most recent constraint sets $\{\mathcal{H}(i)\}_{i=k-L(k)+1}^k$. The algorithm derivation will treat the most general case where $L(k)$ is allowed to vary from one updating instant to another, that is, the case of a variable data reuse factor. Thereafter, we provide algorithm implementations for the case of fixed number of data-reuses (i.e., $L(k) = L$), and the case of $L(k) \leq L_{\text{max}}$ (i.e., $L(k)$ is upper bounded but allowed to vary). The proposed algorithm, SM-PAPA, includes the SM-AP algorithm [17, 29] as a special case and is particularly useful whenever the input signal is highly correlated. As with the SM-PNLMS algorithm, the main idea is to allocate different weights to the filter coefficients using a weighting matrix $G(k)$.

#### 4.1. General algorithm derivation

The SM-PAPA is derived so that its coefficient vector after updating belongs to the set $\psi^{L(k)}(k)$ corresponding to the intersection of $L(k) < N$ past constraint sets, that is,

$$\psi^{L(k)}(k) = \bigcap_{i=k-L(k)+1}^k \mathcal{H}(i).$$

(10)

The number of data-reuses $L(k)$ employed at time instant $k$ is allowed to vary with time. If the previous estimate belongs to the $L(k)$ past constraint sets, that is, $w(k) \in \psi^{L(k)}(k)$, no coefficient update is required. Otherwise, the SM-PAPA performs an update according to the following optimization criterion:

$$w(k + 1) = \arg \min_{w} \|w - w(k)\|^{2}_{G^{-1}(k)}$$

subject to: $d(k) - X^T(k)w^* = p(k),$

(11)

where vector $d(k) \in \mathcal{L}(k)$ contains the desired outputs related to the $L(k)$ last time instants, vector $p(k) \in \mathcal{L}(k)$ has components that obey $|p_i(k)| < \gamma$ and so specify a point
in $\psi^{(k)}(k)$, and matrix $X(k) \in \mathbb{C}^{N \times L(k)}$ contains the corresponding input vectors, that is,
\[
p(k) = [p_1(k)p_2(k) \cdots p_{L(k)}(k)]^T,\]
\[
d(k) = [d(k)d(k-1) \cdots d(k-L(k)+1)]^T,\]
\[
X(k) = [x(k)x(k-1) \cdots x(k-L(k)+1)].\]

Applying the method of Lagrange multipliers for solving the minimization problem of (11), the update equation of the most general SM-PAPA version is obtained as
\[
w(k+1) = \left[ w(k) + G(k)X(k) \left[ X^H(k)G(k)X(k) \right]^{-1} \times \left[ e^*(k) - p^*(k) \right], \right. \]
\[
\left. \begin{array}{ll} 
\times [X^H(k)G(k)X(k)]^{-1} & \text{if } |e(k)| > \gamma \\
\times [X^H(k)G(k)X(k)]^{-1} & \text{otherwise,} \\
\end{array} \right. \]
\[
= \begin{cases} 
\times [X^H(k)G(k)X(k)]^{-1} & \text{otherwise,} \\
\times [X^H(k)G(k)X(k)]^{-1} & \text{if } |e(k)| > \gamma \\
\end{cases} \]
\[
\times [X^H(k)G(k)X(k)]^{-1} a(k)e^*(k)u_1, \] (13)

where $e(k) = d(k) - X^T(k)w^*(k)$. The recursion above requires that matrix $X^H(k)X(k)$, needed for solving the vector of Lagrange multipliers, is nonsingular. To avoid problems, a regularization factor can be included in the inverse (common in conventional AP algorithms), that is, $[X^H(k)X(k) + \delta I]^{-1}$ with $\delta \ll 1$. The choice of $p_i(k)$ can fit each problem at hand.

### 4.2. SM-PAPA with fixed number of data reuses, $L(k) = L$

Following the ideas of [17], a particularly simple SM-PAPA version is obtained if $p_i(k)$ for $i \neq 1$ corresponds to the a posteriori error $e(k-i+1) = d(k-i+1) - w^H(k)x(k-i+1)$ and $p_1(k) = ye(k)/e(k)|$. The simplified SM-PAPA version has recursion given by
\[
w(k+1) = \left[ w(k) + G(k)X(k) \right. \]
\[
\times [X^H(k)G(k)X(k)]^{-1} a(k)e^*(k)u_1, \] (14)

where $u_1 = [10 \cdots 0]^T$ and $a(k)$ is given by (7).

Due to the special solution involving the $L \times 1$ vector $u_1$ in (14), a computationally efficient expression for the coefficient update is obtained by partitioning the input signal matrix as $^3$
\[
X(k) = [x(k)U(k)], \] (15)

where $U(k) = [x(k-1) \cdots x(k-L+1)]$. Substituting the partitioned input matrix in (14) and carrying out the multiplications, we get after some algebraic manipulations (see [9])
\[
w(k+1) = w(k) + \frac{a(k)e^*(k)}{\phi^H(k)G(k)\phi(k)} G(k)\phi(k), \] (16)

where vector $\phi(k)$ is defined as
\[
\phi(k) = x(k) - U(k)[U^H(k)G(k)U(k)]^{-1}U^H(k)G(k)x(k). \] (17)

This representation of the SM-PAPA is computationally attractive as the dimension of the matrix to be inverted is reduced from $L \times L$ to $(L - 1) \times (L - 1)$. As with the SM-PNLMS algorithms, $G(k)$ is a diagonal matrix whose elements are computed according to (9). Algorithm 1 shows the recursions for the SM-PAPA.

The peak computational complexity of the SM-PAPA of Algorithm 1 is similar to that of the conventional PAPA algorithm for the case of unity step size (such that the reduced dimension strategy can be employed). However, one important gain of using the SM-PAPA as well as any other SM algorithm, is the reduced number of computations for those time instants where no updates are required. The lower average complexity due to the sparse updating in time can provide substantial computational savings, that is, lower power consumption. Taking into account that the matrix inversion used in the proposed algorithm needs $O((L - 1)^3)$ complex operations and that $N \gg L$, the cost of the SM-PAPA is $O(NL^2)$ operations per update. Furthermore, the variable data-reuse scheme used by the algorithm proposed in the following, the SM-REDPAPA, reduces even more the computational load by varying the complexity from the SM-PAPA to the SM-PNLMS.

\[
\begin{array}{l}
\text{Algorithm 1: Set-membership proportionate affine-projection algorithm with a fixed number of data reuses.} \\
\end{array}
\]
4.3. SM-PAPA with variable data reuse

For the particular case when the data-reuse factor \( L(k) \) is time varying, the simplified SM-PAPA version in (14) no longer guarantees that the \textit{a posteriori} error is such that \(|e(k-i+1)| \leq \gamma \) for \( i \neq 1 \). This is the case, for example, when the number of data reuses is increased from one update instant to another, that is, \( L(k) > L(k-1) \).

In order to provide an algorithm that belongs to the set \( y^L(k) \) in (10), we can choose the elements of vector \( p(k) \) to be

\[
p_i(k) = \begin{cases} \frac{e(k-i+1)}{|e(k-i+1)|} & \text{if } |e(k-i+1)| > \gamma \\ e(k-i+1) & \text{otherwise} \end{cases}
\]

for \( i = 1, \ldots, L(k) \) with \( e(k) = e(k) \). With the above choice of \( p(k) \), the SM-AP recursions become

\[
w(k+1) = w(k) + G(k)X(k) \\
\times \left[ X^{l_1}(k)G(k)X(k) \right]^{-1} \Lambda^*(k)1_{L(k) 	imes 1},
\]

where matrix \( \Lambda(k) \) is a diagonal matrix whose diagonal elements \( \Lambda(k)_{ii} \) are specified by

\[
[\Lambda(k)]_{ii} = \alpha_i(k)e(k-i+1) \\
= \begin{cases} 1 - \frac{\gamma}{|e(k-i+1)|} & \text{if } |e(k-i+1)| > \gamma \\ e(k-i+1) & \text{otherwise} \end{cases}
\]

and \( 1_{L(k) 	imes 1} = [1, \ldots, 1]^T \).

Another feature of the above algorithm is the possibility to correct previous solutions that for some reason did not satisfy the constraint \(|e(k-i+1)| \leq \gamma \) for \( i \neq 1 \). At this point \(|e(k-i+1)| > \gamma \) for \( i \neq 1 \) could originate from a finite precision implementation or the introduction of a regularization parameter in the inverse in (19).

As can be seen from (20), the amount of zero entries can be significant if \( L(k) \) is large. In Section 5, this fact is exploited in order to obtain a more computationally efficient version of the SM-AP algorithm. Next we consider how to assign a proper data-reuse factor at each time instant.

4.4. Variable data-reuse factor

This section proposes a rule for selecting the number of data-reuses \( L(k) \) to be used at each coefficient update. It can be observed that the main difference in performance between the SM-PAPA and the SM-PNLMS algorithms is in the transient. Generally, the SM-PAPA algorithm has faster convergence than the SM-NLMS algorithm in colored environments. On the other hand, close to the steady state solution, their performances are comparable in terms of excess of MSE. Therefore, a suitable assignment rule increases the data-reuse factor when the solution is far from steady state and reduces to one when close to steady-state (i.e., the SM-PNLMS update).

As discussed previously, \( \alpha_1(k) \) in (20) is a good indicator of how close to steady-state solution is. If \( \alpha_1(k) \to 1 \), the solution is far from the current constraint set which would suggest that the data-reuse factor \( L(k) \) should be increased toward a predefined maximum value \( L_{\text{max}} \). If \( \alpha_1(k) \to 0 \), then \( L(k) \) should approach one resulting in an SM-PNLMS update. Therefore, we propose to use a variable data-reuse factor of the form

\[
L(k) = f(\alpha_1(k)),
\]

where the function \( f(\cdot) \) should satisfy \( f(0) = 1 \) and \( f(1) = L_{\text{max}} \) with \( L_{\text{max}} \) denoting the maximum number of data reuses allowed. In other words, the above expression should quantize \( \alpha_1(k) \) into \( L_{\text{max}} \) regions

\[
I_p = \{ l_{p-1} < \alpha_1(k) \leq l_p \}, \quad p = 1, \ldots, L_{\text{max}}
\]

defined by the decision levels \( l_p \). The variable data-reuse factor is then given by the relation

\[
L(k) = p \quad \text{if } \alpha_1(k) \in I_p.
\]

Indeed, there are many ways in which we could choose the decision variables \( l_p \). In the simulations provided in Section 6, we consider two choices for \( l_p \). The first approach consists of uniformly quantizing \( \alpha_1(k) \) into \( L_{\text{max}} \) regions. The second approach is to use \( l_p = e^{-\beta l_{\text{max}} - p}/l_{\text{max}} \) and \( l_0 = 0 \), where \( \beta \) is a positive constant [29]. This latter choice leads to a variable data-reuse factor on the form

\[
L(k) = \max \left\{ 1, \left[ L_{\text{max}} \left( \frac{1}{\beta} \ln \alpha_1(k) + 1 \right) \right] \right\},
\]

where the operator \( \lfloor \cdot \rfloor \) rounds the element \( \lfloor \cdot \rfloor \) to the nearest integer. Table 1 shows the resulting values of \( \alpha_1(k) \) for both approaches in which the number of reuses should be changed for a maximum of five reuses, usually the most practical case. The values of the decision variables of the second approach provided in the table were calculated with the above expression using \( \beta = 2 \).

5. Reduced complexity version of the variable data-reuse algorithm

This section presents an alternative implementation of the SM-PAPA in (19) that properly reduces the dimensions of the matrices in the recursions.

Assume that, at time instant \( k \), the diagonal of \( \Lambda(k) \) specified by (20) has \( P(k) \) nonzero entries (i.e., \( L(k) = P(k) \) zero
entries. Let $T(k) \in \mathbb{R}^{L(k) \times L(k)}$ denote the permutation matrix that permutes the columns of $X(k)$ such that the resulting input vectors corresponding to nonzero values in $\Lambda(k)$ are shifted to the left, that is, we have
\[
X(k) = X(k)T(k) = [\tilde{X}(k)U(k)],
\]
where matrices $\tilde{X}(k) \in \mathbb{C}^{N \times P(k)}$ and $U(k) \in \mathbb{C}^{N \times [L(k)-P(k)]}$ contain the vectors giving nonzero and zero values on the diagonal of $\Lambda(k)$, respectively. Matrix $T(k)$ is constructed such that the column vectors of matrices $\tilde{X}(k)$ and $U(k)$ are ordered according to their time index.

Using the relation $T(k)^{T} = I_{L(k) \times L(k)}$, we can rewrite the SM-PAPA recursion as
\[
w(k + 1) = w(k) + G(k)X(k)
\]
\[
\times [T(k)T^{T}(k)X^{H}(k)G(k)X(k)T(k)T^{T}(k)]^{-1} \Lambda^{+}(k)I_{L(k) \times 1}
\]
\[
= w(k) + G(k)X(k)
\]
\[
\times [T(k)\tilde{X}^{H}(k)G(k)\tilde{X}(k)T^{T}(k)]^{-1} \Lambda^{+}(k)I_{L(k) \times 1}
\]
\[
= w(k) + G(k)\tilde{X}(k)[\tilde{X}^{H}(k)G(k)\tilde{X}(k)]^{-1} \Lambda^{+}(k),
\]
\]
(26)
\]
where vector $\lambda(k) \in \mathbb{C}^{L(k) \times 1}$ contains the $P(k)$ nonzero adaptive step sizes of $\Lambda(k)$ as the first elements (ordered in time) followed by $L(k) - P(k)$ zero entries, that is,
\[
\lambda(k) = \begin{bmatrix}
\tilde{X}(k) \\
0_{[L(k)-P(k)] \times 1}
\end{bmatrix},
\]
(27)
\]
where the elements of $\tilde{X}(k)$ are the $P(k)$ nonzero adaptive step sizes (ordered in time) of the form $\tilde{X}(k) = (1 - \gamma(\epsilon(k))\epsilon(k))c(k)$.

Due to the special solution involving $\lambda(k)$ in (27), the following computationally efficient expression for the coefficient update is obtained using the partition in (25) (see the appendix)
\[
w(k + 1) = w(k) + G(k)\Phi(k)[\Phi^{H}(k)G(k)\Phi(k)]^{-1} \Lambda^{+}(k),
\]
\]
(28)
\]
where matrix $\Phi(k) \in \mathbb{C}^{N \times P(k)}$ is defined as
\[
\Phi(k) = [\tilde{X}(k) - U(k)[U^{H}(k)G(k)U(k)^{-1}U^{H}(k)G(k)\tilde{X}(k)]],
\]
(29)
\]

This representation of the SM-PAPA is computationally attractive as the dimension of the matrices involved is lower than that of the version described by (19)-(20). Algorithm 2 shows the recursion for the reduced-complexity SM-PAPA, where the $L(k)$ can be chosen as described in the previous section.

### 6. SIMULATION RESULTS

In this section, the performances of the SM-PNLMS algorithm and the SM-PAPA are evaluated in a system identification experiment. The performance of the NLMS, the IPNLMS, the SM-NLMS, and the SM-AP algorithms are also compared.

#### 6.1. Fixed number of data reuses

The first experiment was carried out with an unknown plant with sparse impulse response that consisted of an $N = 50$ truncated FIR model from a digital microwave radio channel.\(^4\) Thereafter, the algorithms were tested for a dispersive channel, where the plant was a complex FIR filter whose co-

\(^4\) The coefficients of this complex-valued baseband channel model can be downloaded from http://spib.rice.edu/spib/microwave.html.
Figure 1: The amplitude of two impulse responses used in the simulations: (a) sparse microwave channel (see Footnote 4), (b) dispersive channel.

efficients were generated randomly. Figure 1 depicts the absolute values of the channel impulse responses used in the simulations. For the simulation experiments, we have used the following parameters: \( \mu = 0.4 \) for the NLMS and the PAP algorithms, \( \gamma = \sqrt{2\sigma_n^2} \) for all SMAF, and \( \kappa = 0.5 \) for all proportionate algorithms. Note that for the IPNLMS and the PAP algorithms, \( g_i(k) = (1 - \kappa)/N + \kappa|w_i(k)/\|w(k)\|_{-1} \) corresponds to the same updating as in [4] when \( \kappa \in [0, 1] \). The parameters were set in order to have fair comparison in terms of final steady-state error. The input signal \( x(k) \) was a complex-valued noise sequence, colored by filtering a zero-mean white complex-valued Gaussian noise sequence \( n_x(k) \) through the fourth-order IIR filter \( x(k) = n_x(k) + 0.95x(k - 1) + 0.19x(k - 2) + 0.09x(k - 3) - 0.5x(k - 4) \), and the SNR was set to 40 dB.

The learning curves shown in Figures 2 and 3 are the result of 500 independent runs and smoothed by a low pass filter. From the learning curves in Figure 2 for the sparse system, it can be seen that the SMF algorithms converge slightly faster than their conventional counterparts to the same level of MSE. In addition to the faster convergence, the SMF algorithms will have a reduced numbers of updates. In 20000 iterations, the number of times an update took place for the SM-PNLMS, the SM-PAPA, and the SM-AP algorithms were 7730 (39%), 6000 (30%), and 6330 (32%), respectively. This should be compared with 20000 updates required by the IPNLMS and PAP algorithms. From Figure 2, we also observe that the proportionate SMF algorithms converge faster than those without proportionate adaptation.

Figure 3 shows the learning curves for the dispersive channel identification, where it can be observed that the performances of the SM-PNLMS and SM-PAPA algorithms are very close to the SM-AP and SM-NLMS algorithms, respectively. In other words, the SM-PNLMS algorithm and the SM-PAPA are not sensitive to the assumption of having a sparse impulse response. In 20000 iterations, the SM-PAPA
and the SM-PNLMS algorithms updated 32% and 50%, respectively, while the SM-AP and SM-NLMS algorithms updated 32% and 49%, respectively.

6.2. Variable data-reuse factor

The SM-PAPA algorithm with variable data-reuse factor was applied to the sparse system example of the previous section. Figures 4 and 5 show the learning curves averaged over 500 simulations for the SM-PAPA for \( L = 2 \) to \( L = 5 \), and SM-REDPAPA for \( L_{\text{max}} = 2 \) to \( L_{\text{max}} = 5 \). Figure 4 shows the results obtained with a uniformly quantized \( \alpha_1(k) \), whereas Figure 5 shows the results obtained using (24) with \( \beta = 2 \). It can be seen that the SM-REDPAPA not only achieves a similar convergence speed, but is also able to reach a lower steady state using fewer updates. The approach of (24) performs slightly better than the one using a uniformly quantized \( \alpha_1(k) \), which slows down during the second part of the transient. On the other hand, the latter approach has the advantage that no parameter tuning is required. Tables 2 and 3 show the number of data-reuses employed for each approach. As can be inferred from the tables, the use of variable data-reuse factor can significantly reduce the overall complexity as compared with the case of keeping it fixed.

7. CONCLUSIONS

This paper presented novel set-membership filtering (SMF) algorithms suitable for applications in sparse environments. The set-membership proportionate NLMS (SM-PNLMS) algorithm and the set-membership proportionate affine projection algorithm (SM-PAPA) were proposed as viable alternatives to the SM-NLMS and SM-AP algorithms. The algorithms benefit from the reduced average computational complexity from the SMF strategy and fast convergence for sparse scenarios resulting from proportionate updating. Simulations were presented for both sparse and dispersive impulse
Table 2: Distribution of the variable data-reuse factor $L(k)$ used in the SM-PAPA for the case when $\alpha_i(k)$ is uniformly quantized.

| $L_{\text{max}}$ | $L(k) = 1$ | $L(k) = 2$ | $L(k) = 3$ | $L(k) = 4$ | $L(k) = 5$ |
|------------------|------------|------------|------------|------------|------------|
| 1                | 100%       | —          | —          | —          | —          |
| 2                | 54.10%     | 45.90%     | —          | —          | —          |
| 3                | 36.55%     | 45.80%     | 17.65%     | —          | —          |
| 4                | 28.80%     | 36.90%     | 26.55%     | 7.75%      | —          |
| 5                | 23.95%     | 29.95%     | 28.45%     | 13.50%     | 4.15%      |

Table 3: Distribution of the variable data-reuse factor $L(k)$ used in the SM-PAPA for the case when $\alpha_i(k)$ is quantized according to (24), $\beta = 2$.

| $L_{\text{max}}$ | $L(k) = 1$ | $L(k) = 2$ | $L(k) = 3$ | $L(k) = 4$ | $L(k) = 5$ |
|------------------|------------|------------|------------|------------|------------|
| 1                | 100%       | —          | —          | —          | —          |
| 2                | 37.90%     | 62.90%     | —          | —          | —          |
| 3                | 28.90%     | 35.45%     | 35.65%     | —          | —          |
| 4                | 28.86%     | 21.37%     | 33.51%     | 18.26%     | —          |
| 5                | 25.71%     | 15.03%     | 23.53%     | 25.82%     | 9.91%      |

responses. It was verified that not only the proposed SMF algorithms can further reduce the computational complexity when compared with their conventional counterparts, the IPNLMS and PAP algorithms, but they also present faster convergence to the same level of MSE when compared with the SM-NLMS and the SM-AP algorithms. The weight assignment of the proposed algorithms utilizes the information provided by a time-varying step size typical for SMF algorithms and is robust to the assumption of sparse impulse response. In order to reduce the overall complexity of the SM-PAPA we proposed to employ a variable data reuse factor. The introduction of a variable data-reuse factor allows significant reduction in the overall complexity as compared to fixed data-reuse factor. Simulations showed that the proposed algorithm could outperform the SM-PAPA with fixed number of data-reuses in terms of computational complexity and final mean-squared error.

**APPENDIX**

The inverse in (26) can be partitioned as

$$[\hat{X}^H(k)G(k)\hat{X}(k)]^{-1} = ([\hat{X}(k)U(k)]^H G(k) [\hat{X}(k)U(k)])^{-1}$$

$$= \begin{bmatrix} A & B^H \\ B & C \end{bmatrix},$$

(A.1)

where

$$A = [\Phi^H(k)G(k)\Phi(k)]^{-1},$$

$$B = -[U(k)^H G(k)U(k)]^{-1} U^H(k)G(k)\hat{X}(k)A,$$  

(A.2)

with $\Phi(k)$ defined as in (29). Therefore,

$$\hat{X}(k) [X^H(k)G(k)\hat{X}(k)]^{-1} X^*(k)$$

$$= \hat{X}(k) \begin{bmatrix} A \\ B \end{bmatrix} X^*(k)$$

$$= \left[\hat{X}(k) - [U^H(k)G(k)U(k)]^{-1} U^H(k)G(k)\hat{X}(k)\right]$$

$$\times [\Phi^H(k)G(k)\Phi(k)]^{-1} X^*(k)$$

$$= \Phi(k) [\Phi^H(k)G(k)\Phi(k)]^{-1} X^*(k).$$

(A.3)

**ACKNOWLEDGMENTS**

The authors would like to thank CAPES, CNPq, FAPERJ (Brazil), and Academy of Finland, Smart and Novel Radios (SMARAD) Center of Excellence (Finland), for partially supporting this work.

**REFERENCES**

[1] R. K. Martin, W. A. Sethares, R. C. Williamson, and C. R. Johnson Jr., “Exploiting sparsity in adaptive filters,” IEEE Transactions on Signal Processing, vol. 50, no. 8, pp. 1883–1894, 2002.

[2] D. L. Duttweiler, “Proportionate normalized least-mean-squares adaptation in echo cancelers,” IEEE Transactions on Speech and Audio Processing, vol. 8, no. 5, pp. 508–518, 2000.

[3] S. L. Gay, “An efficient, fast converging adaptive filter for network echo cancellation,” in Proceedings of the 32nd Asilomar Conference on Signals, Systems & Computers, vol. 1, pp. 394–398, Pacific Grove, Calif, USA, November 1998.

[4] J. Benesty and S. L. Gay, “An improved PNLMS algorithm,” in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’02), vol. 2, pp. 1881–1884, Orlando, Fla, USA, May 2002.

[5] B. D. Rao and B. Song, “Adaptive filtering algorithms for promoting sparsity,” in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ‘03), vol. 6, pp. 361–364, Hong Kong, April 2003.

[6] A. W. H. Khong, J. Benesty, and P. A. Naylor, “An improved proportionate multi-delay block adaptive filter for packet-switched network echo cancellation,” in Proceedings of the 13th European Signal Processing Conference (EUSIPCO ’05), Antalya, Turkey, September 2005.

[7] K. Doğançay and P. Naylor, “Recent advances in partial update and sparse adaptive filters,” in Proceedings of the 13th European Signal Processing Conference (EUSIPCO ’05), Antalya, Turkey, September 2005.

[8] A. Deshpande and S. L. Grant, “A new multi-algorithm approach to sparse system adaptation,” in Proceedings of the 13th European Signal Processing Conference (EUSIPCO ’05), Antalya, Turkey, September 2005.

[9] S. Werner, J. A. Apolinário Jr., P. S. R. Diniz, and T. I. Laakso, “A set-membership approach to normalized proportionate adaptation algorithms,” in Proceedings of the 13th European Signal Processing Conference (EUSIPCO ’05), Antalya, Turkey, September 2005.

[10] H. Deng and M. Dorodováčki, “Proportionate adaptive algorithms for network echo cancellation,” IEEE Transactions on Signal Processing, vol. 54, no. 5, pp. 1794–1803, 2006.
[11] O. Tanrikulu and K. Doğançay, “Selective-partial-update normalized least-mean-square algorithm for network echo cancellation,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’02)*, vol. 2, pp. 1889–1892, Orlando, Fla, USA, May 2002.

[12] J. Kivinen and M. K. Warmuth, “Exponentiated gradient versus gradient descent for linear predictors,” *Information and Computation*, vol. 132, no. 1, pp. 1–63, 1997.

[13] J. Benesty, T. Gansler, D. Morgan, M. Sondhi, and S. Gay, Eds., *Advances in Network and Acoustic Echo Cancellation*, Springer, Boston, Mass, USA, 2001.

[14] O. Hoshuyama, R. A. Goubran, and A. Sugiyama, “A generalized proportionate variable step-size algorithm for fast changing acoustic environments,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’04)*, vol. 4, pp. 161–164, Montreal, Quebec, Canada, May 2004.

[15] S. Gollamudi, S. Nagaraj, S. Kapoor, and Y.-F. Huang, “Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size,” *IEEE Signal Processing Letters*, vol. 5, no. 5, pp. 111–114, 1998.

[16] P. S. R. Diniz and S. Werner, “Set-membership binormalized data-reusing LMS algorithms,” *IEEE Transactions on Signal Processing*, vol. 51, no. 1, pp. 124–134, 2003.

[17] S. Werner and P. S. R. Diniz, “Set-membership affine projection algorithm,” *IEEE Signal Processing Letters*, vol. 8, no. 8, pp. 231–235, 2001.

[18] S. Gollamudi, S. Kapoor, S. Nagaraj, and Y.-F. Huang, “Set-membership adaptive equalization and an updater-shared implementation for multiple channel communications systems,” *IEEE Transactions on Signal Processing*, vol. 46, no. 9, pp. 2372–2385, 1998.

[19] A. V. Malipatil, Y.-F. Huang, S. Andra, and K. Bennett, “Kernelized set-membership approach to nonlinear adaptive filtering,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’05)*, vol. 4, pp. 149–152, Philadelphia, Pa, USA, March 2005.

[20] E. Fogel and Y.-F. Huang, “On the value of information in system identification—bounded noise case,” *Automatica*, vol. 18, no. 2, pp. 229–238, 1982.

[21] S. Dasgupta and Y.-F. Huang, “Asymptotically convergent modified recursive least-squares with data-dependent updating and forgetting factor for systems with bounded noise,” *IEEE Transactions on Information Theory*, vol. 33, no. 3, pp. 383–392, 1987.

[22] J. R. Deller Jr., M. Nayeri, and M. S. Liu, “Unifying the landmark developments in optimal bounding ellipsoid identification,” *International Journal of Adaptive Control and Signal Processing*, vol. 8, no. 1, pp. 43–60, 1994.

[23] D. Joachim and J. R. Deller Jr., “Multiweight optimization in optimal bounding ellipsoid algorithms,” *IEEE Transactions on Signal Processing*, vol. 54, no. 2, pp. 679–690, 2006.

[24] S. Gollamudi, S. Nagaraj, and Y.-F. Huang, “Blind equalization with a deterministic constant modulus cost-a set-membership filtering approach,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’00)*, vol. 5, pp. 2765–2768, Istanbul, Turkey, June 2000.

[25] P. S. R. Diniz and S. Werner, “Set-membership binormalized data-reusing algorithms,” in *Proceedings of the IFAC Symposium on System Identification (SYSID ’00)*, vol. 3, pp. 869–874, Santa Barbara, Calif, USA, June 2000.
Research Article

Wavelet-Based MPNLMS Adaptive Algorithm for Network Echo Cancellation

Hongyang Deng¹ and Milojˇs Doroslovaˇcki²

¹ Freescale Semiconductor, 7700 W. Parmer Lane, Austin, TX 78729, USA
² Department of Electrical and Computer Engineering, The George Washington University, 801 22nd Street, N.W. Washington, DC 20052, USA

Received 30 June 2006; Revised 23 December 2006; Accepted 24 January 2007

The μ-law proportionate normalized least mean square (MPNLMS) algorithm has been proposed recently to solve the slow convergence problem of the proportionate normalized least mean square (PNLMS) algorithm after its initial fast converging period. But for the color input, it may become slow in the case of the big eigenvalue spread of the input signal’s autocorrelation matrix. In this paper, we use the wavelet transform to whiten the input signal. Due to the good time-frequency localization property of the wavelet transform, a sparse impulse response in the time domain is also sparse in the wavelet domain. By applying the MPNLMS technique in the wavelet domain, fast convergence for the color input is observed. Furthermore, we show that some nonsparse impulse responses may become sparse in the wavelet domain. This motivates the usage of the wavelet-based MPNLMS algorithm. Advantages of this approach are documented.

Copyright © 2007 H. Deng and M. Doroslovaˇcki. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

With the development of packet-switching networks and wireless networks, the introduced delay of the echo path increases dramatically, thus entailing a longer adaptive filter. It is well known that long adaptive filter will cause two problems: slow convergence and high computational complexity. Therefore, we need to design new algorithms to speed up the convergence with reasonable computational burden.

Network echo path is sparse in nature. Although the number of coefficients of its impulse response is big, only a small portion has significant values (active coefficients). Others are just zero or unnoticeably small (inactive coefficients). Several algorithms have been proposed to take advantage of the sparseness of the impulse response to achieve faster convergence, lower computational complexity, or both. One of the most popular algorithms is the proportionate normalized least mean square (PNLMS) algorithm [1, 2]. The main idea is assigning different step-size parameters to different coefficients based on their previously estimated magnitudes. The bigger the magnitude, the bigger step-size parameter will be assigned. For a sparse impulse response, most of the coefficients are zero, so most of the update emphasis concentrates on the big coefficients, thus increasing the convergence speed.

The PNLMS algorithm, as demonstrated by several simulations, has very fast initial convergence for sparse impulse response. But after the initial period, it begins to slow down dramatically, even becoming slower than normalized least mean square (NLMS) algorithm. The PNLMS++ [2] algorithm cannot solve this problem although it improves the performance of the PNLMS algorithm.

The μ-law PNLMS (MPNLMS) algorithm proposed in [3–5] uses specially chosen step-size control factors to achieve faster overall convergence. The specially chosen step-size control factors are really an online and causal approximation of the optimal step-size control factors that provide the fastest overall convergence of a proportionate-type steepest descent algorithm. The relationship between this deterministic proportionate-type steepest descent algorithm and proportionate-type NLMS stochastic algorithms is discussed in [6].

In general, the advantage of using the proportionate-type algorithms (PNLMS, MPLMS) is limited to the cases when the input signal is white and the impulse response to be identified is sparse. Now, we will show that we can extend the
advantageous usage of the MPLMS algorithm by using the wavelet transform to cases when the input signal is colored or when the impulse response to be identified is nonsparse.

2. WAVELET DOMAIN MPNLMS

2.1. Color input case

The optimal step-size control factors are derived under the assumption that the input is white. If the input is a color signal, which is often the case for network echo cancellation, the convergence time of each coefficient also depends on the eigenvalues of the input signal’s autocorrelation matrix. Since, in general, we do not know the statistical characteristics of the input signal, it is impossible to derive the optimal step-size control factors without introducing more computational complexity in adaptive algorithm. Furthermore, the big eigenvalue spread of the input signal’s autocorrelation matrix slows down the overall convergence based on the standard LMS performance analysis [7].

One solution of the slow convergence problem of LMS for the color input is the so-called transform domain LMS [7]. By using a unitary transform such as discrete Fourier transform (DFT) and discrete cosine transform (DCT), we can make the input signal’s autocorrelation matrix nearly diagonal. We can further normalize the transformed input vector by the estimated power of each input tap to make the autocorrelation matrix close to the identity matrix, thus decreasing the eigenvalue spread and improving the overall convergence.

But, there is another effect of working in the transform domain: the adaptive filter is now estimating the transform coefficients of the original impulse response [8]. The number of active coefficients to be identified can differ from the number of active coefficients in the original impulse response. In some cases, it can be much smaller and in some cases, it can be much larger.

The MPNLMS algorithm works well only for sparse impulse response. If the impulse response is not sparse, that is, most coefficients are active, the MPNLMS algorithm’s performance degrades greatly. It is well known that if the system is sparse in time domain, it is nonsparse in frequency domain. For example, if a system has only one active coefficient in the time domain (very sparse), all of its coefficients are active in the frequency domain. Therefore, DFT and DCT will transform a sparse impulse response into nonsparse, so we cannot apply the MPNLMS algorithm.

Discrete wavelet transform (DWT) has gained a lot of attention for signal processing in recent years. Due to its good time-frequency localization property, it can transform a time domain sparse system into a sparse wavelet domain system [8]. Let us consider the network echo path illustrated in Figure 1. This is a sparse impulse response. From Figure 2, we see that it is sparse in the wavelet domain, as well. Here, we have used the 9-level Haar wavelet transform on 512 data points. Also, the DWT has the similar band-partitioning property as DFT or DCT to whiten the input signal. Therefore, we can apply the MPNLMS algorithm directly on the transformed input to achieve fast convergence for color input.

The proposed wavelet MPNLMS (WMPNLMS) algorithm is listed in Algorithm 1, where \( x(k) \) is the input signal vector in the time domain, \( L \) is the number of adaptive filter coefficients, \( T \) represents DWT, \( x_T(k) \) is the input signal vector in the wavelet domain, \( x_{T,l}(k) \) is the \( l \)th component of \( x_T(k) \), \( \hat{w}_T(k) \) is the adaptive filter coefficient vector in the wavelet domain, \( \hat{w}_{T,l}(k) \) is the \( l \)th component of \( \hat{w}_T(k) \), \( \hat{y}(k) \) is the output of the adaptive filter, \( d(k) \) is the reference signal, \( e(k) \) is the error signal driving the adaptation, \( \delta_{x_{T,l}}^2(k) \) is the estimated average power of the \( l \)th input tap in the wavelet domain, \( \alpha \) is the forgetting factor with typical value 0.95, \( \beta \) is the step-size parameter, and \( \delta_{p} \) and \( \rho \) are small positive numbers used to prevent the zero or extremely small adaptive
filter coefficients from stalling. The parameter $\varepsilon$ defines the neighborhood boundary of the optimal adaptive filter coefficients. The instant when all adaptive filter coefficients have crossed the boundary defines the convergence time of the adaptive filter. Definition of the matrix $T$ can be found in [9, 10]. Computationally efficient algorithms exist for calculation of $x_T(k)$ due to the convolution-downsampling structure of DWT. The extreme case of computational simplicity corresponds to the usage of the Haar wavelets [11]. The average power of the $i$th input tap in the wavelet domain is estimated recursively by using the exponentially decaying time-window of unit area. There are alternative ways to do the estimation. A common theme in all of them is to find the proper window of unit area. There are alternative ways to do the estimation power of the input signal $x_T(k)$ is calculated as $\sigma_x^2(k) = \sum_{i=1}^{M} g_i(k) y_i(k+1)$.

\[
x(k) = [x(k)x(k-1) \cdots x(k-L+1)]^T
\]
\[
x_T(k) = T x(k)
\]
\[
y_T(k) = x_T^2(k) w_T(k)
\]
\[
e(k) = d(k) - y_T(k)
\]

For $i = 1$ to $L$
\[
\alpha_{i,j}^2(k) = \alpha \alpha_{i,j}^2(k-1) + (1-\alpha)x^2_{i,j}(k)
\]
End
\[
D(k+1) = \text{diag}[\alpha_{1,1}^2(k), \ldots, \alpha_{2,L}^2(k)]
\]
\[
\omega_T(k+1) = \omega_T(k) + \beta D^{-1}(k+1) G(k+1) x_T(k) e(k)
\]
\[
G(k+1) = \text{diag}[g_1(k+1), \ldots, g_L(k+1)]
\]
\[
F(\hat{w}(k)) = \ln(1+\mu |\hat{w}(k)|), \quad 1 \leq l \leq L, \quad \mu = 1/\varepsilon
\]
\[
y_{\min}(k+1) = \rho \max_{i} \{ \delta_{\min}, F(|\hat{w}_i(k)|), \ldots, F(|\hat{w}_L(k)|) \}
\]
\[
y_i(k+1) = \max_{i} \{ y_{\min}(k+1), F(|\hat{w}_i(k)|) \}
\]
\[
g_i(k+1) = \frac{y_i(k+1)}{(1/L) \sum_{i=1}^{L} y_i(k+1)}, \quad 1 \leq l \leq L.
\]

Algorithm 1: WMPNLMS algorithm.

In some networks, nonsparse impulse responses can appear. Figure 4 shows an echo path impulse response of a digital subscriber line (DSL) system. We can see that it is not sparse in the time domain. It has a very short fast changing segment and a very long slow decreasing tail [11]. If we apply the MPNLMS algorithm on this type of impulse response, we cannot expect that we will improve the convergence speed. But if we transform the impulse response into wavelet domain by using the 9-level Haar wavelet transform, it turns into a sparse impulse response as shown in Figure 5. Now, the WMPNLMS can speed up the convergence.

To evaluate the performance of the WMPNLMS algorithm identifying the DSL echo path shown in Figure 4, we use an adaptive filter with 512 taps. The input signal is white. As previously, we use $\delta_p = 0.01$, $\rho = 0.01$, and $\beta$ that provides the same steady-state error as the NLMS, MPNLMS, and SPNLMS algorithms. Figure 6 shows learning curves for identifying the DSL echo path. We can see that the NLMS algorithm and the wavelet-based NLMS algorithm have nearly the same performance, because the input signal is white. The MPNLMS algorithm has marginal improvement in this case because the impulse response of the DSL echo path is not very sparse. But the WMPNLMS algorithm has much faster
convergence due to the sparseness of the impulse response in the wavelet domain and the algorithm’s proportionate adaptation mechanism. The wavelet-based NLMS algorithm also identifies a sparse impulse response, but does not speed up the convergence by using the proportionate adaptation mechanism. Compared to the computational and memory requirements listed in [5, Table IV] for the MPNLMS algorithm, the WMPNLMS algorithm, in the case of Haar wavelets with $M$ levels of decomposition, requires $M + 2L$ more multiplications, $L − 1$ more divisions, $2M + L − 1$ more additions/subtractions, and $2L − 1$ more memory elements.

Figure 4: DSL echo path impulse response.

Figure 5: Wavelet domain coefficients for DSL echo path impulse response in Figure 4.

Figure 6: Learning curves for identifying DSL network echo path.

3. CONCLUSION

We have shown that by applying the MPNLMS algorithm in the wavelet domain, we can improve the convergence of the adaptive filter identifying an echo path for the color input. Essential for the good performance of the WMPNLMS is that the wavelet transform preserve the sparseness of the echo path impulse response after the transformation. Furthermore, we have shown that by using the WMPNLMS, we can improve convergence for certain nonsparse impulse responses, as well. This happens since the wavelet transform converts them into sparse ones.

REFERENCES

[1] D. L. Duttweiler, “Proportionate normalized least-mean-squares adaptation in echo cancelers,” IEEE Transactions on Speech and Audio Processing, vol. 8, no. 5, pp. 508–518, 2000.
[2] S. L. Gay, “An efficient, fast converging adaptive filter for network echo cancellation,” in Proceedings of the 32nd Asilomar Conference on Signals, Systems & Computers (ACSSC ’98), vol. 1, pp. 394–398, Pacific Grove, Calif, USA, November 1998.
[3] H. Deng and M. Doroslovački, “Modified PNLMS adaptive algorithm for sparse echo path estimation,” in Proceedings of the Conference on Information Sciences and Systems, pp. 1072–1077, Princeton, NJ, USA, March 2004.
[4] H. Deng and M. Doroslovački, “Improving convergence of the PNLMS algorithm for sparse impulse response identification,” IEEE Signal Processing Letters, vol. 12, no. 3, pp. 181–184, 2005.
[5] H. Deng and M. Doroslovački, “Proportionate adaptive algorithms for network echo cancellation,” IEEE Transactions on Signal Processing, vol. 54, no. 5, pp. 1794–1803, 2006.
[6] M. Doroslovački and H. Deng, “On convergence of proportionate-type NLMS adaptive algorithms,” in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’06), vol. 3, pp. 105–108, Toulouse, France, May 2006.

[7] S. Haykin, Adaptive Filter Theory, Prentice-Hall, Upper Saddle River, NJ, USA, 4th edition, 2002.

[8] M. Doroslovački and H. Fan, “Wavelet-based linear system modeling and adaptive filtering,” IEEE Transactions on Signal Processing, vol. 44, no. 5, pp. 1156–1167, 1996.

[9] G. Strang and T. Nguyen, Wavelets and Filter Banks, Wellesley-Cambridge Press, Wellesley, Mass, USA, 1996.

[10] M. Shamma and M. Doroslovački, “Comparison of wavelet and other transform based LMS adaptive algorithms for colored inputs,” in Proceedings of the Conference on Information Sciences and Systems, vol. 2, pp. FPS 17–FPS 20, Princeton, NJ, USA, March 2000.

[11] M. Doroslovački and H. Fan, “On-line identification of echo-path impulse responses by Haar-wavelet-based adaptive filter,” in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’95), vol. 2, pp. 1065–1068, Detroit, Mich, USA, May 1995.
A Low Delay and Fast Converging Improved Proportionate Algorithm for Sparse System Identification

Andy W. H. Khong, 1 Patrick A. Naylor, 1 and Jacob Benesty 2

1 Department of Electrical and Electronic Engineering, Imperial College London, Exhibition Road, London SW7 2AZ, UK
2 INRS-EMT, Université du Québec, Suite 6900, 800 de la Gauchetière Ouest, Montréal, QC, Canada H5A 1K6

Received 4 July 2006; Revised 1 December 2006; Accepted 24 January 2007

Recommended by Kutluyil Dogancay

A sparse system identification algorithm for network echo cancellation is presented. This new approach exploits both the fast convergence of the improved proportionate normalized least mean square (IPNLMS) algorithm and the efficient implementation of the multidelay adaptive filtering (MDF) algorithm inheriting the beneficial properties of both. The proposed IPMDF algorithm is evaluated using impulse responses with various degrees of sparseness. Simulation results are also presented for both speech and white Gaussian noise input sequences. It has been shown that the IPMDF algorithm outperforms the MDF and IPNLMS algorithms for both sparse and dispersive echo path impulse responses. Computational complexity of the proposed algorithm is also discussed.

Copyright © 2007 Andy W. H. Khong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Research on network echo cancellation is increasingly important with the advent of voice over internet protocol (VoIP). In such systems where traditional telephony equipment is connected to the packet-switched network, the echo path impulse response, which is typically of length 64–128 milliseconds, exhibits an “active” region in the range of 8–12 milliseconds duration and consequently, the impulse response is dominated by regions where magnitudes are close to zero making the impulse response sparse. The “inactive” region is due to the presence of bulk delay caused by network propagation, encoding, and jitter buffer delays [1]. Other applications for sparse system identification include wavelet identification using marine seismic signals [2] and geophysical seismic applications [3, 4].

Classical adaptive algorithms with a uniform step-size across all filter coefficients such as the normalized least mean square (NLMS) algorithm have slow convergence in sparse network echo cancellation applications. One of the first algorithms which exploits the sparse nature of network impulse responses is the proportionate normalized least mean square (PNLMS) algorithm [5] where each filter coefficient is updated with an independent step-size which is proportional to the estimated filter coefficient. Subsequent improved versions such as the IPNLMS [6] and IIPNLMS [7] algorithms were proposed, which achieve improved convergence by introducing a controlled mixture of proportionate (PNLMS) and nonproportionate (NLMS) adaptation. Consequently, these algorithms perform better than PNLS for sparse and, in some cases, for dispersive impulse responses. To reduce the computational complexity of PNLMS, the sparse partial update NLMS (SPNLMS) algorithm was proposed [8] where, similar to the selective partial update NLMS (SPUNLMS) algorithm [9], only taps corresponding to the $M$ largest absolute values of the product of input signal and filter coefficients are selected for adaptation. An optimal step-size for PNLS has been derived in [10] and employing an approximate $\mu$-law function, the proposed segment PNLS (SPNLMS) outperforms the PNLS algorithm.

In recent years, frequency-domain adaptive algorithms have become popular due to their efficient implementation. These algorithms incorporate block updating strategies whereby the fast Fourier transform (FFT) algorithm [11] is used together with the overlap-save method [12, 13]. One of the main drawbacks of these approaches is the delay introduced between the input and output which can be equivalent to the length of the adaptive filter. Consequently, for long impulse responses, this delay can be considerable since the number of filter coefficients can be several thousands [14]. To
mitigate this problem, Soo and Pang proposed the multidelay filtering (MDF) algorithm [15] which uses a block length $N$ independent of the filter length $L$. Although it has been well-known, from the computational complexity point of view, that $N = L$ is the optimal choice, the MDF algorithm nevertheless is more efficient than time-domain implementations even for $N < L$ [16].

In this paper, we propose and evaluate the improved proportionate multidelay filtering (IPMDF) algorithm for sparse impulse responses. The IPMDF algorithm exploits both the improvement in convergence brought about by the proportionality control of the IPNLMS algorithm and the efficient implementation of the MDF structure. As will be explained, direct extension of the IPNLMS algorithm to the MDF structure is inappropriate due to the dimension mismatch between the update vectors. Consequently, in contrast to the MDF structure, adaptation for the IPMDF algorithm is performed in the time domain. We then evaluate the performance of IPMDF using impulse responses with various degrees of sparseness [18, 19]. This paper is organized as follows. In Section 2, we review the PNLMS, IPNLMS, and MDF algorithms. We then derive the proposed IPMDF algorithm in Section 3 while Section 3.2 presents the computational complexity. Section 4 shows simulation results and Section 5 concludes our work.

2. ADAPTIVE ALGORITHMS FOR SPARSE SYSTEM IDENTIFICATION

With reference to Figure 1, we first define filter coefficients and tap-input vector as

$$\hat{h}(n) = [\hat{h}_0(n), \hat{h}_1(n), \ldots, \hat{h}_{L-1}(n)]^T, \quad x(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^T,$$

where $L$ is the adaptive filter length and the superscript $T$ is defined as the transposition operator. The adaptive filter will model the unknown impulse response $h(n)$ using the near-end signal

$$y(n) = x^T(n)h(n) + v(n) + w(n),$$

where $v(n)$ and $w(n)$ are defined as the near-end speech signal and ambient noise, respectively. For simplicity, we will temporarily ignore the effects of double talk and ambient noise, that is, $v(n) = w(n) = 0$, in the description of algorithms.

2.1. The PNLMS and IPNLMS algorithms

The proportionate normalized least mean square (PNLMS) [5] and improved proportionate normalized least mean square (IPNLMS) [6] algorithms have been proposed for network echo cancellation where the impulse response of the system is sparse. These algorithms can be generalized using the following set of equations:

$$e(n) = y(n) - \hat{h}^T(n-1)x(n),$$

$$\hat{h}(n) = \hat{h}(n-1) + \frac{\mu Q(n-1) x(n) e(n)}{x^T(n) Q(n-1) x(n) + \delta},$$

$$Q(n-1) = \text{diag} \{q_0(n-1), \ldots, q_{L-1}(n-1)\},$$

where $\mu$ is the adaptive step-size and $\delta$ is the regularization parameter. The $L \times L$ diagonal control matrix $Q(n)$ determines the step-size of each filter coefficient and is dependent on the specific algorithm as described below.

2.1.1. PNLMS

The PNLMS algorithm assigns higher step-sizes for coefficients with higher magnitude using a control matrix $Q(n)$. Elements of the control matrix for PNLMS can be expressed as [5]

$$q_i(n) = \frac{\kappa_i(n)}{\sum_{i=0}^{L-1} \kappa_i(n)},$$

$$\kappa_i(n) = \max \{\rho \times \max \{|y|, |\hat{h}_0(n)|, \ldots, |\hat{h}_{L-1}(n)|, |\hat{h}_l(n)|\} \}$$

(6)

with $l = 0, 1, \ldots, L - 1$ being the tap-indices. The parameter $y$, with a typical value of 0.01, prevents $\hat{h}(n)$ from stalling during initialization stage where $\hat{h}(0) = 0_{L \times 1}$ while $\rho$ prevents coefficients from stalling when they are much smaller than the largest coefficient. The regularization parameter $\delta$ in (4) for PNLMS should be taken as

$$\delta_{\text{PNLMS}} = \frac{\delta_{\text{NLMS}}}{L},$$

(7)

where $\delta_{\text{NLMS}} = \sigma^2_x$ is the variance of the input signal [6]. It can be seen that for $\rho \geq 1$, PNLMS is equivalent to NLMS.

---

1 An earlier version of this work was presented at the EUSIPCO 2005 special session on sparse and partial update adaptive filters [17].
2.1.2. IPNLMS

An enhancement of PNLMS is the IPNLMS algorithm [6] which is a combination of PNLMS and NLMS with the relative significance of each controlled by a factor $\alpha$. The elements of the control matrix $Q(n)$ for IPNLMS are given by

$$q_i(n) = \frac{1 - \alpha}{2L} + (1 + \alpha) \frac{|h_i(n)|}{2\|h\|_1 + \epsilon}, \quad (8)$$

where $\epsilon$ is a small value and $\| \cdot \|_1$ is the $l_1$-norm operator. It can be seen from the second term of (8) that the magnitude of the estimated taps is normalized by the $l_1$ norm of $\hat{h}$. This shows that the weighting on the step-size for IPNLMS is dependent only on the relative scaling of the filter coefficients as opposed to their absolute values. Results presented in [6, 17] have shown that good choices of $\alpha$ values are 0, 0.5, and 0.75. The regularization parameter $\delta$ in (4) for IPNLMS should be taken [6] as

$$\delta_{\text{IPNLMS}} = \frac{1 - \alpha}{2L} \delta_{\text{NLMS}}. \quad (9)$$

This choice of regularization ensures that the IPNLMS algorithm achieves the same asymptotic steady-state normalized misalignment compared to that of the NLMS algorithm. It can be seen that IPNLMS is equivalent to NLMS when $\alpha = -1$ while, for $\alpha$ close to 1, IPNLMS behaves like PNLMS.

2.2. The frequency-domain MDF algorithm

Frequency-domain adaptive filtering has been introduced as a form of improving the efficiency of time-domain algorithms. Although substantial computational savings can be achieved, one of the main drawbacks of frequency-domain approaches is the inherent delay introduced [13]. The multi-delay filtering (MDF) algorithm [15] was proposed to mitigate the delay problem by partitioning the adaptive filter into $K$ blocks each having length $N$ such that $L = KN$. The MDF algorithm can be summarized by first letting $m$ be the frame index and defining the following quantities:

$$x(mN) = [x(mN), \ldots, x(mN - L + 1)]^T, \quad (10)$$

$$X(m) = [x(mN), \ldots, x(mN + N - 1)], \quad (11)$$

$$y(m) = [y(mN), \ldots, y(mN + N - 1)]^T, \quad \hat{y}(m) = [\hat{y}(mN), \ldots, \hat{y}(mN + N - 1)]^T = X^T(m)\hat{h}(m), \quad (12)$$

$$\hat{Y}(m) = \hat{y}(mN), \ldots, \hat{y}(mN + N - 1)]^T = \hat{X}^T(m)\hat{h}(m), \quad (13)$$

$$e(m) = y(m) - \hat{y}(m) = [e(mN), \ldots, e(mN + N - 1)]^T. \quad (14)$$

We note that $X(m)$ is a Toeplitz matrix of dimension $L \times N$. Defining $k$ as the block index and $T(m - k)$ as an $N \times N$ Toeplitz matrix such that

$$T(m-k) = \begin{bmatrix}
    x(mN - kN) & \cdots & x(mN - kN - N + 1) \\
    x(mN - kN + 1) & \ddots & \vdots \\
    \vdots & \ddots & \vdots \\
    x(mN - kN + N - 1) & \cdots & x(mN - kN)
\end{bmatrix}, \quad (15)$$

it can be shown using (13) and (15) that the adaptive filter output can be expressed as

$$\hat{y}(m) = \sum_{k=0}^{K-1} T(m-k)\hat{h}_k(m), \quad (16)$$

where

$$\hat{h}_k(m) = [\hat{h}_{kN}(m), \hat{h}_{kN+1}(m), \ldots, \hat{h}_{kN+N-1}(m)]^T \quad (17)$$

is the $k$th subfilter of $h(m)$ for $k = 0, 1, \ldots, K - 1$.

It can be shown that the Toeplitz matrix $T(m - k)$ can be transformed, by doubling its size, to a circulant matrix

$$C(m - k) = \begin{bmatrix}
    T'(m-k) & T(m-k) \\
    T(m-k) & T'(m-k)
\end{bmatrix} \quad (18)$$

with

$$T'(m-k) = \begin{bmatrix}
    x(mN - kN + N) & \cdots & x(mN - kN + 1) \\
    x(mN - kN - 1) & \ddots & \vdots \\
    \vdots & \ddots & \vdots \\
    x(mN - kN - N + 1) & \cdots & x(mN - kN)
\end{bmatrix}. \quad (19)$$

The resultant circulant matrix $C$ can then be decomposed [20] as

$$C = F^{-1}DF, \quad (20)$$

where $F$ is a $2N \times 2N$ Fourier matrix and $D$ is a diagonal matrix whose elements are the discrete Fourier transform of the first column of $C$. Note that the diagonal of $T'$ is arbitrary, but it is normally equal to the first sample of the previous block $k - 1$ [16]. We now define the frequency-domain quantities:

$$\gamma(m) = F \begin{bmatrix}
    0_{N \times 1} \\
    y(m)
\end{bmatrix}, \quad \hat{h}_k(m) = F \begin{bmatrix}
    \hat{h}_k(m) \\
    0_{N \times 1}
\end{bmatrix}, \quad (21)$$

$$e(m) = F \begin{bmatrix}
    0_{N \times 1} \\
    e(m)
\end{bmatrix}, \quad G^{00} = FW^{00}F^{-1},$$

$$W^{01} = \begin{bmatrix}
    0_{N \times N} & 0_{N \times N} \\
    0_{N \times N} & I_{N \times N}
\end{bmatrix}, \quad G^{10} = FW^{01}F^{-1},$$

$$W^{10} = \begin{bmatrix}
    I_{N \times N} & 0_{N \times N} \\
    0_{N \times N} & 0_{N \times N}
\end{bmatrix}. \quad (21)$$
The MDF adaptive algorithm is then given by the following equations:

\[
ge(m) = y(m) - G^{01} \sum_{k=0}^{K-1} D(m-k) \hat{h}_k(m-1),
\]

\[
S_{\text{MDF}}(m) = \lambda S_{\text{MDF}}(m-1) + (1 - \lambda) D^*(m) D(m),
\]

\[
\hat{h}_k(m) = \hat{h}_k(m-1) + \mu G^{10} D^*(m-k) \times [S_{\text{MDF}}(m) + \delta_{\text{MDF}}]^{-1} e(m),
\]

where \( \ast \) denotes complex conjugate, \( 0 \leq \lambda < 1 \) is the forgetting factor, and \( \mu = \beta (1 - \lambda) \) is the step-size with \( 0 < \beta \leq 1 \) [16]. It has been found through simulation that this value of \( \mu \) exhibits stability in terms of convergence for speech signals. Letting \( \sigma_x^2 \) be the input signal variance, the initial regularization parameters [16] are \( S_{\text{MDF}}(0) = \sigma_x^2 / 100 \) and \( \delta_{\text{MDF}} = 20 \sigma_x^2 N / L \). For a nonstationary signal, \( \sigma_x^2 \) can be estimated in a piecewise manner at each iteration by \( \hat{x}_2^2(n) = (n)x(n)/(2N) \) where \( x(n) \) is the first column of the \( 2N \times 2N \) matrix \( C \). Convergence analysis for the MDF algorithm is provided in [21].

3. THE IPMDF ALGORITHM

3.1. Algorithmic formulation

The proposed IPMDF algorithm exploits both the fast convergence of the improved proportionate normalized least mean square (IPPNLMS) algorithm and the efficient implementation of the multidelay adaptive filtering (MDF) algorithm inheriting the beneficial properties of both. We note that direct use of \( Q(n) \), with elements as described by (8), into the weight update equation in (24) is inappropriate since the former is in the time domain whereas the latter is in the frequency domain. Thus our proposed method will be to update the filter coefficients in the time domain. This is achieved by first defining the matrices

\[
\tilde{W}^{10} = \begin{bmatrix} I_{N \times N} & 0_{N \times N} \end{bmatrix}, \quad \tilde{G}^{10} = \tilde{W}^{10} F^{-1}.
\]

We next define, for \( k = 0, 1, \ldots, K-1 \),

\[
q_k(m) = \begin{bmatrix} q_{kN}(m), q_{kN+1}(m), \ldots, q_{kN+N-1}(m) \end{bmatrix}
\]

as the partitioned control elements of the \( k \)th block such that each element in this block is now determined by

\[
q_{kN+j}(m) = \frac{1 - \alpha}{2L} + (1 + \alpha) \frac{||\hat{h}_{kN+j}(m)||}{2||\hat{h}||_1 + \epsilon},
\]

where \( k = 0, 1, \ldots, K-1 \) is the block index while \( j = 0, 1, \ldots, N-1 \) is the tap-index of each \( k \)th block. The IPMDF algorithm update equation is then given by

\[
\hat{h}_k(m) = \hat{h}_k(m-1) + L \mu q_k(m) \tilde{G}^{10} D^*(m-k) \times [S_{\text{IPMDF}}(m) + \delta_{\text{IPMDF}}]^{-1} e(m),
\]

where the diagonal control matrix \( Q_k(m) = \text{diag}(q_k(m)) \). The proposed IPMDF algorithm performs updates in the time domain by first computing the gradient of the adaptive algorithm given by \( D^*(m-k)[S_{\text{IPMDF}}(m) + \delta_{\text{IPMDF}}]^{-1} e(m) \) in the frequency domain. The matrix \( \tilde{G}^{10} \) then converts this gradient to the time domain so that multiplication with the (time-domain) control matrix \( Q_k(m) \) is possible. The estimated impulse response \( \hat{h}_k(m) \) is then transformed into the frequency domain for error computation given by

\[
e(m) = y(m) - G^{01} \sum_{k=0}^{K-1} D(m-k) \hat{h}_k(m-1).
\]

The IPMDF algorithm can be summarized as shown in Algorithm 1.

3.2. Computational complexity

We consider the computational complexity of the proposed IPMDF algorithm. We note that although the IPMDF algorithm is updated in the time domain, the error \( e(m) \) is generated using frequency-domain coefficients and hence five FFT-blocks are required. Since a 2N point FFT requires 2N log \( N \) real multiplications, the number of multiplications required per output sample for each algorithm is
described by the following relations:

\[
\begin{align*}
\text{IPNLMS:} & \quad 4L, \\
\text{FLMS:} & \quad 8 + 10 \log_2 L, \\
\text{MDF:} & \quad 8K + (4K + 6) \log_2 N, \\
\text{IPMDF:} & \quad 10K + (4K + 6) \log_2 N.
\end{align*}
\]

(30)

It can be seen that the complexity of IPMDF is only modestly higher than MDF. However, as we will see in Section 4, the performance of IPMDF far exceeds that of MDF for both speech and white Gaussian noise (WGN) inputs.

4. RESULTS AND DISCUSSIONS

The performance of IPMDF is compared with MDF and IPNLMS in the context of network echo cancellation. This performance can be quantified using the normalized misalignment defined by

\[
\eta(m) = \frac{\|h - \hat{h}(m)\|_2^2}{\|h\|_2^2},
\]

(31)

where \(\| \cdot \|_2^2\) is defined as the squared \(l_2\)-norm operator. Throughout our simulations, we assume that the length of the adaptive filter is equivalent to that of the unknown system. Results are presented over a single trial and the following parameters are chosen for all simulations:

\[
\alpha = -0.75, \\
\lambda = \left[1 - \frac{1}{(3L)}\right]^N, \\
\beta = 1, \\
\mu = \beta \times (1 - \lambda), \\
S_{\text{MDF}}(0) = \frac{\sigma_x^2}{100}, \\
\delta_{\text{MDF}} = \frac{\sigma_x^2 20N}{L}, \\
S_{\text{IPMDF}}(0) = \frac{(1 - \alpha)\sigma_x^2}{200}, \\
\delta_{\text{IPMDF}} = \frac{20(1 - \alpha)\sigma_x^2 N}{(2L)}, \\
\delta_{\text{NLMS}} = \sigma_x^2, \\
\delta_{\text{IPNLMS}} = \frac{1 - \alpha}{2L} \delta_{\text{NLMS}}.
\]

(32)

These choices of parameters allow algorithms to converge to the same asymptotic value of \(\eta(m)\) for fair comparison.

4.1. Recorded impulse responses

In this first experiment, we investigate the variation of the rate of convergence with frame size \(N\) for IPMDF using an impulse response of a 64 milliseconds network hybrid recorded at 8 kHz sampling frequency as shown in Figure 2. Figure 3 shows the convergence with various frame sizes \(N\) for IPMDF using a white Gaussian noise (WGN) input sequence. An uncorrelated WGN sequence \(w(n)\) is added to achieve a signal-to-noise ratio (SNR) of 30 dB. It can be seen that the convergence is faster for smaller \(N\) since the adaptive filter coefficients are being updated more frequently. Additional simulations for \(N < 64\) have indicated that no further significant improvement in convergence performance is obtained for lower \(N\) values.

We compare the relative rate of convergence of the IP-MDF, MDF, IPNLMS, and NLMS algorithms using the same impulse response. As before, \(w(n)\) is added to achieve an SNR of 30 dB. The frame size for IPMDF and MDF was chosen to be \(N = 64\) while the step-size of IPNLMS and NLMS was adjusted so that its final misalignment is the same as that for IPMDF and MDF. This corresponds to \(\mu_{\text{IPNLMS}} = \mu_{\text{NLMS}} = 0.15\). Figure 4 shows the convergence for the respective algorithms using a WGN sequence. It can be seen that there is a significant improvement in normalized misalignment of approximately 5 dB during convergence for the IPMDF compared to MDF and IPNLMS.
We compare the tracking performance of the algorithms as shown in Figure 5 using a WGN input sequence. In this simulation, an echo path change, comprising an additional 12-sample delay, was introduced after 3 seconds. As before, the frame size for the IPMDF and MDF algorithms is $N = 64$ while for IPNLMS and NLMS, $\mu_{\text{IPNLMS}} = \mu_{\text{NLMS}} = 0.15$ is used. We see that IPMDF achieves the highest initial rate of convergence. When compared with MDF, the IPMDF algorithm has a higher tracking capability following the echo path change at 3 seconds. Compared with the IPNLMS algorithm, a delay is introduced by block processing the data input for both the MDF and IPMDF algorithms. As a result, IPNLMS achieves a better tracking capability than the MDF algorithm. The tracking capability of NLMS is slower compared to IPNLMS and IPMDF due to its relatively slow convergence rate. Although delay exists for the IPMDF algorithm, the reduction in delay due to the multidelay structure allows the IPMDF algorithm to achieve an improvement of 2 dB over IPNLMS after echo path change.

Figure 6 compares the convergence performance of IPNLMS, IPMDF, and MDF using the same experimental setup as before but using a speech input from a male speaker. An echo path change, comprising an additional 12-sample delay, is introduced at 16 seconds. It can be seen that IPMDF achieves approximately 5 dB improvement in normalized misalignment during initial convergence compared to the MDF algorithm.

### 4.2. Synthetic impulse responses with various degrees of sparseness

We illustrate the robustness of IPMDF to impulse response sparseness. Impulse responses with various degrees of sparseness are generated synthetically using an $L \times 1$ exponentially decaying window [18] which is defined as

$$u = [p_1 e^{-1/\psi}, e^{-2/\psi}, \ldots, e^{-(L_u-1)/\psi}]^T,$$

where the $L_p \times 1$ vector $p$ models the bulk delay and is a zero mean WGN sequence with variance $\sigma_p^2$ and $L_u = L - L_p$ is the length of the decaying window while $\psi \in \mathbb{Z}^+$ is the decay constant. Defining an $L_u \times 1$ vector $b$ as a zero mean WGN sequence with variance $\sigma_b^2$, the $L \times 1$ synthetic impulse response can then be expressed as

$$B = \text{diag}(b), \quad h = \begin{bmatrix} I_{L_p} \times I_{L_p} & 0_{L_p} \times L_u & B \end{bmatrix} u. \quad (34)$$

The sparseness of an impulse response can be quantified using the sparseness measure [18, 19]

$$\xi(h) = \frac{L}{L - \sqrt{L} \left(1 - \frac{\|h\|_1}{\sqrt{L}\|h\|_2}\right)}, \quad (35)$$

It has been shown in [18] that $\xi(h)$ reduces with $\psi$. Figure 7 shows an illustrative example set of impulse responses generated using (34) with $\sigma_p^2 = 1.055 \times 10^{-4}$, $\sigma_b^2 = 0.9146$,
$L = 512$, and $L_p = 64$. These impulse responses with various degrees of sparseness were generated using decay constants (a) $\psi = 10$, (b) $\psi = 50$, (c) $\psi = 150$, and (d) $\psi = 300$ giving sparseness measures of (a) $\xi = 0.8767$, (b) $\xi = 0.6735$, (c) $\xi = 0.4216$, and (d) $\xi = 0.3063$, respectively. We now investigate the performance of IPNLMS, MDF, and IPMDF using white Gaussian noise input sequences for impulse responses generated using $0.3 \leq \xi \leq 0.9$ as controlled by $\psi$. As before, $w(n)$ is added to achieve an SNR of 30 dB. Figure 8 shows the variation in time to reach $\eta(m) = -20$ dB normalized misalignment with sparseness measure $\xi$ controlled using exponential window $\psi$. Due to the proportional control of step-sizes, significant increase in the rate of convergence for IPNLMS and IPMDF can be seen as the sparseness of the impulse responses increases for high $\xi$. For all cases of sparseness, the IPMDF algorithm exhibits the highest rate of convergence compared to IPNLMS and MDF hence demonstrating the robustness of IPMDF to the sparse nature of the unknown system.

**Figure 7:** Impulse responses controlled using (a) $\psi = 10$, (b) $\psi = 50$, (c) $\psi = 150$, and (d) $\psi = 300$ giving sparseness measure (a) $\xi = 0.8767$, (b) $\xi = 0.6735$, (c) $\xi = 0.4216$, and (d) $\xi = 0.3063$.

**Figure 8:** Time to reach $-20$ dB ($T_{20}$) normalized misalignment for (a) IPNLMS, (b) MDF and (c) IPMDF algorithms with sparseness measure $\xi$ controlled using exponential decay factor $\psi$. 

---

**Figure 8:** Time to reach $-20$ dB ($T_{20}$) normalized misalignment for (a) IPNLMS, (b) MDF and (c) IPMDF algorithms with sparseness measure $\xi$ controlled using exponential decay factor $\psi$. 

---
5. CONCLUSION

We have proposed the IPMDF algorithm for echo cancellation with sparse impulse responses. This algorithm exploits both the improvement in convergence brought about by the proportionality control of IPNLMS and the efficient implementation in the frequency domain of MDF. Simulation results, using both WGN and speech inputs, have shown that the improvement in initial convergence and tracking of IP-MDF over MDF for both sparse and dispersive impulse responses far outweighs the modest increase in computational cost.

REFERENCES

[1] J. Radecki, Z. Zilic, and K. Radecka, “Echo cancellation in IP networks,” in Proceedings of the 45th Midwest Symposium on Circuits and Systems, vol. 2, pp. 219–222, Tulsa, Okla., USA, August 2002.
[2] M. Boujida and J.-M. Boucher, “Higher order statistics applied to wavelet identification of marine seismic signals,” in Proceedings of European Signal Processing Conference (EUSIPCO ’96), Trieste, Italy, September 1996.
[3] Y.-F. Cheng and D. M. Etter, “Analysis of an adaptive technique for modeling sparse systems,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, no. 2, pp. 254–264, 1989.
[4] E. A. Robinson and T. S. Durrani, Geophysical Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, USA, 1986.
[5] D. L. Duttweiler, “Proportionate normalized least-mean-squares adaptation in echo cancellers,” IEEE Transactions on Speech and Audio Processing, vol. 8, no. 5, pp. 508–518, 2000.
[6] J. Benesty and S. L. Gay, “An improved PNLMS algorithm,” in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’02), vol. 2, pp. 1881–1884, Orlando, Fla., USA, May 2002.
[7] J. Cui, P. A. Naylor, and D. T. Brown, “An improved IPNLMS algorithm for echo cancellation in packet-switched networks,” in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’04), vol. 4, pp. 141–144, Montreal, Que., Canada, May 2004.
[8] H. Deng and M. Doroslovaˇcki, “New sparse adaptive algorithms using partial update,” in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’04), vol. 2, pp. 845–848, Montreal, Que., Canada, May 2004.
[9] K. Doganacay and O. Tanrikulu, “Adaptive filtering algorithms with selective partial updates,” IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 48, no. 8, pp. 762–769, 2001.
[10] H. Deng and M. Doroslovaˇcki, “Improving convergence of the PNLMS algorithm for sparse impulse response identification,” IEEE Signal Processing Letters, vol. 12, no. 3, pp. 181–184, 2005.
[11] J. W. Cooley and J. W. Tukey, “An algorithm for the machine calculation of complex Fourier series,” Mathematics of Computation, vol. 19, no. 90, pp. 297–301, 1965.
[12] S. Haykin, Adaptive Filter Theory, Information and System Science Series, Prentice-Hall, Englewood Cliffs, NJ, USA, 4th edition, 2002.
[13] J. J. Shynk, “Frequency-domain and multirate adaptive filtering,” IEEE Signal Processing Magazine, vol. 9, no. 1, pp. 14–37, 1992.
[14] E. Hänsl and G. U. Schmidt, “Hands-free telephones - joint control of echo cancellation and postfiltering,” Signal Processing, vol. 80, no. 11, pp. 2295–2305, 2000.
[15] J.-S. Soo and K. K. Pang, “Multidelay block frequency domain adaptive filter,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, no. 2, pp. 373–376, 1990.
[16] J. Benesty, T. Gänzl, D. R. Morgan, M. M. Sondhi, and S. L. Gay, Advances in Network and Acoustic Echo Cancellation, Springer, New York, NY, USA, 2001.
[17] A. W. H. Khong, J. Benesty, and P. A. Naylor, “An improved proportionate multi-delay block adaptive filter for packet-switched network echo cancellation,” in Proceedings of the 13th European Signal Processing Conference (EUSIPCO ’05), Antalya, Turkey, September 2005.
[18] J. Benesty, Y. A. Huang, J. Chen, and P. A. Naylor, “Adaptive algorithms for the identification of sparse impulse responses,” in Selected Methods for Acoustic Echo and Noise Control, E. Hänsl and G. Schmidt, Eds., chapter 5, pp. 125–153, Springer, New York, NY, USA, 2006.
[19] P. O. Hoyer, “Non-negative matrix factorization with sparseness constraints,” Journal of Machine Learning Research, vol. 5, pp. 1457–1469, 2004.
[20] R. Gray, “On the asymptotic eigenvalue distribution of toepitz matrices,” IEEE Transactions on Information Theory, vol. 18, no. 6, pp. 725–730, 1972.
[21] J. Lee and S.-C. Chong, “On the convergence properties of multidelay frequency domain adaptive filter,” in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’99), vol. 4, pp. 1865–1868, Phoenix, Ariz., USA, March 1999.
Research Article

Analysis of Transient and Steady-State Behavior of a Multichannel Filtered-\(\times\) Partial-Error Affine Projection Algorithm

Alberto Carini\(^1\) and Giovanni L. Sicuranza\(^2\)

\(^1\)Information Science and Technology Institute, University of Urbino “Carlo Bo”, 61029 Urbino, Italy
\(^2\)Department of Electrical, Electronic and Computer Engineering, University of Trieste, 34127 Trieste, Italy

Received 28 April 2006; Revised 24 November 2006; Accepted 27 November 2006

Recommended by Kutluyil Dogancay

The paper provides an analysis of the transient and the steady-state behavior of a filtered-\(\times\) partial-error affine projection algorithm suitable for multichannel active noise control. The analysis relies on energy conservation arguments, it does not apply the independence theory nor does it impose any restriction to the signal distributions. The paper shows that the partial-error filtered-\(\times\) affine projection algorithm in presence of stationary input signals converges to a cyclostationary process, that is, the mean value of the coefficient vector, the mean-square error and the mean-square deviation tend to periodic functions of the sample time.

Copyright © 2007 A. Carini and G. L. Sicuranza. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Active noise controllers are based on the destructive interference in given locations of the noise produced by some primary sources and the interfering signals generated by some secondary sources driven by an adaptive controller [1]. A commonly used strategy is based on the so-called feed-forward methods, where some reference signals measured in the proximity of the noise source are available. These signals are used together with the error signals captured in the proximity of the zone to be silenced in order to adapt the controller. Single-channel and multichannel schemes have been proposed in the literature according to the number of reference sensors, error sensors, and secondary sources used. A single-channel active noise controller makes use of a single reference sensor, actuator, and error sensor and it gives, in principle, attenuation of the undesired disturbance in the proximity of the point where the error sensor is located. In the multichannel approach, in order to spatially extend the silenced region, multiple reference sensors, actuators and error sensors are used. Due to the multiplicity of the signals involved, to the strong correlations between them and to the long impulse response of the acoustic paths, multichannel active noise controllers suffer the complexity of the coefficient updates, the data storage requirements, and the slow convergence of the adaptive algorithms [2]. To improve the convergence speed, different filtered-\(\times\) affine projection (FX-AP) algorithms have been used [3, 4] in place of the usual filtered-\(\times\) LMS algorithms, but at the expense of a further, even though limited, increment of the complexity of updates. Various techniques have been proposed in the literature to keep low the implementation complexity of adaptive FIR filters having long impulse responses. Most of them can be usefully applied to the filtered-\(\times\) algorithms, too, especially in the multichannel situations. A first approach is based on the so-called interpolated FIR filters [5], where a few impulse response samples are removed and then their values are derived using some type of interpolation scheme. However, the success of this implementation is based on the hypothesis that practical FIR filters have an impulse response with a smooth predictable envelope, which is not applicable to the acoustic paths. Another approach is based on data-selective updates which are sparse in time. This approach can be suitably described in the framework of the set-membership filtering (SMF) where a filter is designed to achieve a specified bound on the magnitude of the output error [6]. Finally, a set of well-established techniques is based on selective partial updates (PU) where selected blocks of filter coefficients are updated at every iteration in a sequential or periodic manner [7] or by using an appropriate selection criterion [8]. Among
the partial update strategies, a simple yet effective approach is provided by the partial error (PE) technique, which has been first applied in [7] for reducing the complexity of linear multichannel controllers equipped with the filtered-x LMS algorithm. The PE technique consists in using sequentially at each iteration only one of the $K$ error sensor signals in place of their combination and it is capable to reduce the adaptation complexity with a factor $K$. In [9], the PE technique was applied, together with other methods, for reducing the computational load of multichannel active noise controllers equipped with filtered-x affine projection (AP) algorithms. When dealing with novel adaptive filters, it is important to assess their performance not only through extensive simulations but also with theoretical analysis results. In the literature, very few results deal with the analysis of filtered-x, affine projection or partial-update algorithms. The convergence analysis results for these algorithms are often based on the independence theory (IT) and they constrain the probability distribution of the input signal to be Gaussian or spherically invariant [10]. The IT hypothesis assumes statistical independence of time-lagged input data vectors. As it is too strong for filtered-x LMS [11] and AP algorithms [12], different approaches have been studied in the literature in order to overcome this hypothesis. In [11], an analysis of the mean weight behavior of the filtered-x LMS algorithm, based only on neglecting the correlation between coefficient and signal vectors, is presented. Moreover, the analysis of [11] does not impose any restriction on the signal distributions. Another analysis approach that avoids IT is applied in [12] for the mean-square performance analysis of AP algorithms. This relies on energy conservation arguments, and no restriction is imposed on the signal distributions. In [4], we applied and adapted the approach of [12] for analyzing the convergence behavior of multichannel FX-AP algorithms. In this paper, we extend the analysis approach of [4] and study the transient and steady-state behavior of a filtered-x partial error affine projection (FX-PE-AP) algorithm. The paper shows that the FX-PE-AP algorithm in presence of stationary input signals converges to a cyclostationary process, that is, that the mean value of the coefficient vector, the mean-square-error, and the mean-square-deviation tend to periodic functions of the sample time. We also show the FX-PE-AP algorithm is capable to reduce the adaptation complexity with a factor $K$ with respect to an approximate FX-AP algorithm introduced in [4], but it also reduces the convergence speed by the same factor.

The paper is organized as follows. Section 2 reviews the multichannel feedforward active noise controller structure and introduces the FX-PE-AP algorithm. Section 3 discusses the asymptotic solution of the FX-PE-AP algorithm and compares it with that of FX-AP algorithms and with the minimum-mean-square solution of the ANC problem. Section 4 presents the analysis of the transient and steady-state behavior of the FX-PE-AP algorithm. Section 5 provides some experimental results. Conclusions follow in Section 6.

Throughout this paper, small boldface letters are used to denote vectors and bold capital letters are used to denote matrices, for example, $\mathbf{x}$ and $\mathbf{X}$, all vectors are column vectors, the boldface symbol $I$ indicates an identity matrix of appropriate dimensions, the symbol $\odot$ denotes linear convolution, $\text{diag}(\cdots)$ is a block-diagonal matrix of the entries, $E[\cdot]$ denotes mathematical expectation, $\parallel \cdot \parallel_2$ is the weighted Euclidean norm, for example, $\| \mathbf{w} \|_2^2 = \mathbf{w}^T \Sigma \mathbf{w}$ with $\Sigma$ a symmetric positive definite matrix, $\text{vec}(\cdot)$ indicates the vector operator and $\text{vec}^{-1}(\cdot)$ the inverse vector operator that returns a square matrix from an input vector of appropriate dimensions, $\otimes$ denotes the Kronecker product, $a\%b$ is the remainder of the division of $a$ by $b$, and $|a|$ is the absolute value of $a$.

2. THE PARTIAL-ERROR FILTERED-\(x\) AP ALGORITHM

The schematic description of a multichannel feedforward active noise controller (ANC) is provided in Figure 1. $I$ reference sensors collect the corresponding input signals from the noise sources and $K$ error sensors collect the error signals at the interference locations. The signals coming from these sensors are used by the controller in order to adaptively estimate $J$ output signals which feed $J$ actuators. The corresponding block diagram is reported in Figure 2. The propagation of the original noise up to the region to be silenced is described by the transfer functions $p_{k,i}(z)$ representing the primary paths. The secondary noise signals propagate through secondary paths, which are characterized by the transfer functions $s_{k,j}(z)$. We assume there is no feedback between loudspeakers and reference sensors. The primary source signals filtered by the impulse responses of the secondary paths model, with transfer functions $\tilde{s}_{k,j}(z)$, are used for the adaptive filter update, and for this reason the adaptation algorithm is called filtered-x. Figure 2 illustrates also the delay-compensation scheme [13] that is used throughout the paper. To compensate for the propagation delay introduced by the secondary paths, the output of the primary paths $d(n)$ is estimated with $\tilde{d}(n)$ by subtracting the output of the secondary paths model from the error sensors signals $d(n)$, and the error signal $e(n)$ between $d(n)$ and the output of the adaptive filter is used for the adaptation of the filter $\mathbf{w}(n)$. A copy of this filter is used for the actuators’ output estimation.

Preliminary and independent evaluations of the secondary paths transfer functions are needed. For generality purposes, the theoretical results we present assume imperfect modelling of the secondary paths (we consider $\tilde{s}_{k,j}(z) \neq s_{k,j}(z)$ for any choice of $j$ and $k$), but all the results hold also for perfect modelling (i.e., for $\tilde{s}_{k,j}(z) = s_{k,j}(z)$). Indeed, the experimental results of Section 5 refer to ANC systems with perfect modelling of the secondary paths. When necessary, we will highlight in the paper the different behavior of the system under perfect and imperfect estimations of the secondary paths.

Very mild assumptions are posed in this paper on the adaptive controller. Indeed, we assume that any input $i$ of the controller is connected to any output $j$ through a filter whose output depends linearly on the filter coefficients, that is, we assume that the $j$th actuator output is given by the following...
vector equation:

\[ y_j(n) = \sum_{i=1}^{I} x_i^T(n) w_{ji}(n), \]  

where \( w_{ji}(n) \) is the coefficient vector of the filter that connects the input \( i \) to the output \( j \) of the adaptive controller, and \( x_i(n) \) is the \( i \)th primary source input signal vector. In particular, \( x_i(n) \) is here expressed as a vector function of the signal samples \( x_i(n) \) whose general form is given by

\[ x_i(n) = [f_1[x_i(n)], f_2[x_i(n)], \ldots, f_N[x_i(n)]]^T, \]  

where \( f_i[\cdot] \), for any \( i = 1, \ldots, N \), is a time-invariant functional of its argument. Equations (1) and (2) include linear filters, truncated Volterra filters of any order \( p \) [14], radial basis function networks [15], filters based on functional expansions [16], and other nonlinear filter structures. In Section 5 we provide experimental results for linear filters, where the vector \( x_i(n) \) reduces to

\[ x_i(n) = [x_i(n), x_i(n-1), \ldots, x_i(n-N+1)]^T, \]  

and for filters based on a piecewise linear functional expansion with the vector \( x_i(n) \) given by

\[ x_i(n) = [x_i(n), x_i(n-1), \ldots, x_i(n-N+1), |x_i(n) - a|, \ldots, |x_i(n-N+1) - a|]^T, \]  

where \( a \) is an appropriate constant.
To introduce the PE-FX-AP algorithm analyzed in subsequent sections, we make use of quantities defined in Table 1.

Our objective is to estimate the coefficient vector \( \mathbf{w}_o = [\mathbf{w}_1^T, \mathbf{w}_2^T, \ldots, \mathbf{w}_J^T]^T \) that minimizes the cost function given in

\[
J_o = E \left[ \sum_{k=1}^{K} \left( d_k(n) + \sum_{j=1}^{J} s_{k,j}(n) \odot (\mathbf{w}_j^T \mathbf{x}(n)) \right)^2 \right].
\] (5)

Several adaptive filters have been proposed in the literature to estimate the filter \( \mathbf{w}_o \). In [4], we have analyzed the convergence properties of the approximate FX-AP algorithm with adaptation rule given by

\[
\mathbf{w}(n + 1) = \mathbf{w}(n) - \mu \sum_{k=0}^{K} \tilde{U}_k(n) \tilde{R}_k^{-1}(n) \tilde{e}_k(n),
\] (6)

where

\[
\tilde{R}_k(n) = \tilde{U}_k^T(n) \tilde{U}_k(n) + \delta \mathbf{I}.
\] (7)

In this paper, we consider the FX-PE-AP algorithm characterized by the adaptation rule of

\[
\mathbf{w}(n + 1) = \mathbf{w}(n) - \mu \tilde{U}_{n \% K}(n) \tilde{R}_{n \% K}^{-1}(n) \tilde{e}_{n \% K}(n),
\] (8)

where \( n \% K \) is the remainder of the division of \( n \) by \( K \). The adaptation rule in (8) has been obtained by applying the PE methodology to the approximate FX-AP algorithm of (6). At each iteration, only one of the \( K \) active error sensor signals is used for the controller adaptation. The error sensor signal employed for the adaptation is chosen with a round-robin strategy. Thus, compared with (6), the FX-PE-AP adaptation in (8) reduces the computational load by a factor \( K \).

The exact value of the estimated residual error \( \tilde{e}_k(n) \) is given by

\[
\tilde{e}_k(n) = d_k(n) + \sum_{j=1}^{J} (s_{k,j}(n) - \tilde{s}_{k,j}(n)) \odot (\mathbf{w}_j^T \mathbf{x}(n))
+ \sum_{j=1}^{J} \mathbf{w}_j^T(n) \tilde{u}_{k,j}(n).
\] (9)

In order to analyze the FX-PE-AP algorithm, we introduce in (9) the approximation

\[
\sum_{j=1}^{J} (s_{k,j}(n) - \tilde{s}_{k,j}(n)) \odot (\mathbf{w}_j^T \mathbf{x}(n)) \equiv \sum_{j=1}^{J} \mathbf{w}_j^T(n) \cdot ((s_{k,j}(n) - \tilde{s}_{k,j}(n)) \odot \mathbf{x}(n)),
\] (10)

which allows us to simplify (9) and to obtain

\[
\tilde{e}_k(n) = d_k(n) + \sum_{j=1}^{J} \mathbf{w}_j^T(n) \tilde{u}_{k,j}(n).
\] (11)

Note that the expression in (11) is correct when we perfectly estimate the secondary paths or when \( \mathbf{w}(n) \) is constant, that is, when we work with small step-size values. On the contrary, the expression in (11) is only an approximation for large step-sizes and in presence of secondary path estimation errors, but it allows an insightful analysis of the effects of these estimation errors.

By introducing the result of (11) in (8), we obtain the following equation:

\[
\mathbf{w}(n + 1) = \mathbf{w}(n) - \mu \tilde{U}_{n \% K}(n) \tilde{R}_{n \% K}^{-1}(n) \sum_{i=0}^{K-1} \tilde{v}_i(n) + \mathbf{m}_i(n),
\] (12)

where

\[
\begin{align*}
\mathbf{w}(n) &= \mathbf{V}_{n \% K}(n) \mathbf{w}(n) - \mathbf{v}_{n \% K}(n), \\
\mathbf{V}_i(n) &= \mathbf{I} - \mu \tilde{U}_i(n) \tilde{R}_i^{-1}(n) \tilde{U}_i(n) + \mathbf{m}_i(n),
\end{align*}
\] (13)

By iterating \( K \) times (13) from \( n = mK + i \) till \( n = mK + i + K - 1 \), with \( m \in \mathbb{N} \) and \( 0 \leq i < K \), we obtain the expression of (15), which will be used for the algorithm analysis,

\[
\mathbf{w}(mK + i + K) = \mathbf{M}_i(mK + i) \mathbf{w}(mK + i) - \mathbf{m}_i(mK + i),
\] (15)

where

\[
\begin{align*}
\mathbf{M}_i(n) &= \mathbf{V}_{(i+K-1) \% K}(n + K - 1) \mathbf{V}_{(i+K-2) \% K}(n + K - 2) \\
&\quad \times \cdots \mathbf{V}_{i \% K}(n),
\end{align*}
\] (16)

\[
\begin{align*}
\mathbf{m}_i(n) &= \mathbf{V}_{(i+K-1) \% K}(n + K - 1) \cdots \mathbf{V}_{(i+1) \% K}(n + 1) \mathbf{v}_{i \% K}(n) \\
&\quad + \mathbf{V}_{(i+K-1) \% K}(n + K - 1) \cdots \mathbf{V}_{(i+2) \% K}(n + 2) \\
&\quad \times \mathbf{v}_{(i+1) \% K}(n + 1) \\
&\quad \cdots \mathbf{v}_{(i+K-1) \% K}(n + K - 1).
\end{align*}
\] (17)

3. THE ASYMPTOTIC SOLUTION

For \( i \) ranging from 0 to \( K - 1 \), (15) provides a set of \( K \) independent equations that can be separately studied. The system matrix \( \mathbf{M}_i(n) \) and excitation matrix \( \mathbf{m}_i(n) \) have different statistical properties for different indexes \( i \). For every \( i \), the recursion in (15) converges to a different asymptotic coefficient vector and it provides different values of the steady-state mean-square error and the mean-square deviation. If the input signals are stationary and if the recursion in (15) is convergent for every \( i \), it can be shown that the algorithm converges to a cyclostationary process of periodicity \( K \).

For every index \( i \), the coefficient vector \( \mathbf{w}(mK + i) \) tends for \( m \to +\infty \) to an asymptotic vector \( \mathbf{w}_{\infty,i} \), which depends on the statistical properties of the input signals. In fact, by taking the expectation of (15) and considering the fixed point of this equation, it can be easily deduced that

\[
\mathbf{w}_{\infty,i} = (E[\mathbf{M}_i(n)] - \mathbf{I})^{-1}E[\mathbf{m}_i(n)].
\] (18)
Since the matrices $E[M_i(n)]$ and $[m_i(n)]$ vary with $i$, so do the asymptotic coefficient vectors $w_{\infty,i}$. Thus, the vector $w(n)$ for $n \rightarrow +\infty$ tends to the periodic sequence formed by the repetition of the $K$ vectors $w_{\infty,i}$ with $i = 0, 1, \ldots, K - 1$.

The asymptotic sequence varies with the step-size $\mu$ and with the estimation errors $\tilde{s}_{k,i}(z) - s_{k,i}(z)$ of the secondary paths. As we already observed for FX-AP algorithms [4], the asymptotic solution in (18) differs from the minimum-mean-square (MMS) solution of the active noise control problem, which is given by (19) [17],

$$w_o = -R_{ud}R_{ud}^{-1}$$  \hspace{0.5cm} (19)
4. TRANSIENT ANALYSIS AND STEADY-STATE ANALYSIS

The transient analysis aims to study the time evolution of the expectation of the weighted Euclidean norm of the coefficient vector $E[\| w(mK + i) \|^2_2] = w(n)^T \Sigma w(n)$ for some choices of the symmetric positive definite matrix $\Sigma$ [12]. Moreover, the limit for $n \rightarrow +\infty$ of the same quantity, again for some appropriate choices of the matrix $\Sigma$, is needed for the steady-state analysis. For simplicity, in the following we assume to work with stationary input signals and, according to (15), we separately analyze the evolution of $E[\| w(mK + i) \|^2_2]$ for the different indexes $i$.

4.1. Energy conservation relation

We first derive a recursive relation for $\| w(mK + i) \|^2_2$. By substituting the expression of (15) in the definition of $\| w(mK + i + K) \|^2_2$, we obtain the relation of

$$
\| w(mK + i + K) \|^2_2 = w^T(mK + i + K) \Sigma w(mK + i + K) - 2w^T(mK + i)q_{\Sigma,i}(mK + i) + m_i^T(mK + i) \Sigma m_i(mK + i),
$$

(23)

where we have introduced the quantities $\Sigma'_i(n)$ and $q_{\Sigma,i}(n)$ which are defined, respectively, in

$$
\Sigma'_i(n) = M_i^T(n) \Sigma M_i(n),
$$

$$
q_{\Sigma,i}(n) = M_i^T(n) \Sigma m_i(n).
$$

(24)

Equation (23) provides an energy conservation relation, which is the basis of our analysis. The relation of (23) has the same role of the energy conservation relation employed in [12]. No approximation has been used for deriving the expression of (23).

4.2. Transient analysis

We are now interested in studying the time evolution of $E[\| w(mK + i) \|^2_2]$ where $\Sigma$ is a symmetric and positive definite square matrix. For this purpose, we follow the approach of [12, 18, 19].

In the analysis of filtered-x and AP algorithms, it is common to assume $w(n)$ to be uncorrelated with some functions of the filtered input signal [11, 12]. This assumption provides good results and is weaker than the hypothesis of the independence theory, which requires the statistical independence of time-lagged input data vectors.

Therefore, in what follows, we introduce the following approximation.

(A1) For every $i$ with $0 \leq i < K$ and for $m \in \mathbb{N}$, we assume $w(mK + i)$ to be uncorrelated with $M_i(mK + i)$ and with $q_{\Sigma,i}(mK + i)$.

In the appendix, we prove the following theorem that describes the transient behavior of the FX-PE-AP algorithm.

**Theorem 1.** Under the assumption (A1), the transient behavior of the FX-PE-AP algorithm with updating rule given by (15) is described by the state recursions

$$
E[w(mK + i + K)] = M_i E[w(mK + i)] - m_i,
$$

$$
W_i(mK + i + K) = G_i W_i(mK + i) + y_i(mK + i),
$$

(25)
where
\[
\begin{align*}
M_i &= E[M_i(n)], \\
m_i &= E[m_i(n)], \\
G_i &= 
\begin{bmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1 \\
-k_{0,i} & -p_{1,i} & -p_{2,i} & \cdots & -p_{M-1,i}
\end{bmatrix}, \\
W_i(n) &= \begin{bmatrix}
E[||w(n)||_{vec^{-1}(\sigma)}] \\
E[||w(n)||_{vec^{-1}(\Sigma)}] \\
\vdots \\
E[||w(n)||_{vec^{-1}(P^{M-1}_{\sigma})}]
\end{bmatrix}, \\
y_i(n) &= \begin{bmatrix}
(g_i^T - 2E[w^T(n)]Q_i)\sigma \\
(g_i^T - 2E[w^T(n)]Q_i)F_i\sigma \\
\vdots \\
(g_i^T - 2E[w^T(n)]Q_i)P^{M-1}_{\sigma}\sigma
\end{bmatrix}
\end{align*}
\]

the $M^2 \times M^2$ matrix $F_i = E[M_i^T(n) \otimes M_i^T(n)$, the $M \times M^2$ matrix $Q_i = E[m_i^T(n) \otimes m_i^T(n)$, the $M^2 \times 1$ vector $g_i = vec\{E[m_i(n)m_i^T(n)\}$, the $p_{ij}$ are the coefficients of the characteristic polynomial of $F_i$, that is, $p_i(x) = x^{M^2} + p_{M-1}x^{M^2-1} + \cdots + p_1x + p_0 = det(xF_i - I)$, and $\sigma = vec\{\Sigma\}$.

Note that since the input signals are stationary, $M_i$, $m_i$, $G_i$, $F_i$, $Q_i$, and $g_i$ are time-independent. On the contrary, $y_i(n)$ depends from the time sample $n$ through $E[w(n)]$.

According to Theorem 1, for every index $i$ the transient behavior of the FX-PE-AP algorithm is described by the cascade of two linear systems, with system matrices $M_i$ and $G_i$, respectively. The stability in the mean sense and in the mean-square sense can be deduced by the stability properties of these two linear systems. Indeed, the FX-PE-AP algorithm will converge in the mean for any step-size $\mu$ such that for every $i$, $|\lambda_{\text{max}}(M_i)| < 1$. The algorithm will converge in the mean-square sense if, in addition, for every $i$ it is $|\lambda_{\text{max}}(G_i)| < 1$.

It should be noted that the matrices $M_i$ and $F_i$ are non-symmetric for both perfect and imperfect secondary path estimates. Thus, the algorithm could originate an oscillatory convergence behavior.

### 4.3. Steady-state behavior

We are here interested in the estimation of the mean-square error (MSE) and the mean-square deviation (MSD) at steady state. The adaptation rule of (15) provides different values of MSE and MSD for the different indexes $i$. Therefore, in what follows, we define

\[
\text{MSD}_i = \lim_{m \to +\infty} E[||w(mK + i) - w_{\infty,i}||^2] = \lim_{m \to +\infty} E[w^T(mK + i)w(mK + i) - ||w_{\infty,i}||^2],
\]

\[
\text{MSE}_i = \lim_{m \to +\infty} E[\sum_{k=1}^{n} c_i^2(mK + i)].
\]

Note that the definition of the MSD in (27) refers to the asymptotic solution $w_{\infty,i}$ instead of the mean-square solution $w_i$ as in [11, 12, 20]. We adopt the definition in (27) because when $\mu$ tends to zero, also the MSD in (27) converges to zero, that is, $\lim_{\mu \to 0} \text{MSE}_i = 0$ for all $i$.

Similar to [4], we make use of the following hypothesis:

(A2) We assume $w(n)$ to be uncorrelated with $\sum_{k=1}^{K} u_i(n) \times u_i^T(n)$ and with $\sum_{k=1}^{K} d_k(n) u_i(n)$.

By exploiting the hypothesis in (A2), the MSE can be expressed as

\[
\text{MSE}_i = S_d + 2R_{uw,i} w_{\infty,i} + \lim_{m \to +\infty} E[w^T(mK + i)R_{uw}w(mK + i)],
\]

where

\[
S_d = E\left[\sum_{k=1}^{K} d_k^2(n)\right],
\]

and $R_{uu}$ and $R_{ud}$ are defined in (20), respectively.

The computations in (27) and (29) require the evaluation of $\lim_{m \to +\infty} E[||w(mK + i)||_{\Sigma}^2]$, where $\Sigma = I$ in (27) and $\Sigma = R_{uu}$ in (29). This limit can be estimated with the same methodology of [12].

If we assume the convergence of the algorithm, when $m \to +\infty$, the recursion in (A.1) becomes

\[
\lim_{m \to +\infty} E[||w(mK + i)||^2_{vec^{-1}(\sigma)}] = \lim_{m \to +\infty} E[||w(mK + i)||^2_{vec^{-1}(\Sigma)}] - 2w_{\infty,i}^T Q_i \sigma + g_i^T \sigma,
\]

which is equivalent to

\[
\lim_{m \to +\infty} E[||w(mK + i)||^2_{vec^{-1}(I - F_i)\sigma}] = -2w_{\infty,i}^T Q_i \sigma + g_i^T \sigma.
\]
For simplicity, we provide results only for a perfect estimate and the transfer functions of the secondary paths are can be evaluated as in

$$\text{MSE}_i = S_d + 2R_{\text{ss}}^T w_{\text{ss}, i} + (g_i^T - 2w_{\text{ss}, i}Q_i)(I - F_i)^{-1} \text{vec} \{R_{\text{ss}}\}. \quad (33)$$

To estimate the MSD, we have to choose \( \sigma \) such that \((I - F_i)\sigma = \text{vec} \{R_{\text{ss}}\}\), that is, \(\sigma = (I - F_i)^{-1} \text{vec} \{R_{\text{ss}}\}\). Therefore, the MSD can be evaluated as in

$$\text{MSD}_i = (g_i^T - 2w_{\text{ss}, i}Q_i)(I - F_i)^{-1} \text{vec} \{I\} - ||w_{\text{ss}, i}||^2. \quad (34)$$

### 5. Experimental Results

In this section, we provide a few experimental results that compare theoretically predicted values with values obtained from simulations.

We first considered a multichannel active noise controller with \( I = 1, J = 2, K = 2 \). The transfer functions of the primary paths are given by

$$p_{11}(z) = 1.0z^{-2} - 0.3z^{-3} + 0.2z^{-4},$$
$$p_{21}(z) = 1.0z^{-2} - 0.2z^{-3} + 0.1z^{-4}, \quad (35)$$

and the transfer functions of the secondary paths are

$$s_{11}(z) = 2.0z^{-1} - 0.5z^{-2} + 0.1z^{-3},$$
$$s_{12}(z) = 2.0z^{-1} - 0.3z^{-2} - 0.1z^{-3},$$
$$s_{21}(z) = 1.0z^{-1} - 0.7z^{-2} - 0.2z^{-3},$$
$$s_{22}(z) = 1.0z^{-1} - 0.2z^{-2} + 0.2z^{-3}. \quad (36)$$

For simplicity, we provide results only for a perfect estimate of the secondary paths, that is, we consider \( \tilde{s}_{ij}(z) = s_{ij}(z) \). The input signal is the normalized logistic noise, which has been generated by scaling the signal \( \xi(n) \) obtained from the logistic recursion \( \xi(n + 1) = \lambda \xi(n)(1 - \xi(n)) \), with \( \lambda = 4 \) and \( \xi(0) = 0.9 \), and by adding a white Gaussian noise to get a 30 dB signal-to-noise ratio. It has been proven for single-channel active noise controllers that in presence of a nonminimum phase secondary path, the controller acts as a predictor of the reference signal and that a nonlinear controller can better estimate a non-Gaussian noise process [15, 21]. In the case of our multichannel active noise controller, the exact solution of the multichannel ANC problem requires the inversion of the \( 2 \times 2 \) matrix \( S \) formed with the transfer functions \( s_{ij} \). The inverse matrix \( S^{-1} \) is formed by IIR transfer functions whose poles are given by the roots of the determinant of \( S \). It is easy to verify that in our example, there is a root outside the unit circle. Thus, also in our case the controller acts as a predictor of the input signal and a nonlinear controller can better estimate the logistic noise. Therefore, in what follows, we provide results for (1) the two-channel linear controller with memory length \( N = 8 \) and (2) the two-channel nonlinear controller with memory length \( N = 4 \) whose input data vector is given in (4), with the constant \( a \) set to 1. Note that despite the two controllers have different memory lengths, they have the same total number of coefficients, that is, \( M = 16 \). In all the experiments, a zero mean, white Gaussian noise, uncorrelated between the microphones, has been added to the error microphone signals \( d_k(n) \) to get a 40 dB signal-to-noise ratio and the parameter \( \delta \) was set to 0.001.

Table 2 and 3 provide with three-digits precision the first eight coefficients of the MMS solution, \( w_0 \), and of the asymptotic solutions of the FX-PE-AP algorithm at even samples, \( w_{0,1} \), and odd samples, \( w_{0,2} \), and of the approximate FX-AP algorithm of (6), \( w_0 \), for \( \mu = 1.0 \) and for the AP orders \( L = 1, 2, \) and 3. Table 2 refers to the linear controller and Table 3 to the nonlinear controller, respectively. From Tables 2 and 3, it is evident that the asymptotic vector varies with the AP order and that the asymptotic solutions \( w_{0,1}, w_{0,2} \), and \( w_0 \) are different. However, we must point out that their difference reduces with the step-size, and for smaller step-sizes it can be difficulty appreciated.

Figure 3 diagrams the steady-state MSE, estimated with (33) or obtained from simulations with time averages over ten million samples, versus step-size \( \mu \) and for AP orders \( L = 1, 2, \) and 3. Similarly, Figure 4 diagrams the steady-state MSE, estimated with (34) or obtained from simulations with time averages over ten million samples. From Figures 3 and 4, we see that the expressions in (33) and in (34) provide accurate estimates of the steady-state MSE and of the steady-state
Table 3: First eight coefficients of the MMS solution \( \mathbf{w}_o \) and of the asymptotic solutions of FX-PE-AP \( \mathbf{w}_{\infty,0}, \mathbf{w}_{\infty,1} \) and of FX-AP algorithm \( \mathbf{w}_\infty \) with the nonlinear controller.

| \( L \) | \( \mathbf{w}_o \) | \( \mathbf{w}_{\infty,0} \) | \( \mathbf{w}_{\infty,1} \) | \( \mathbf{w}_\infty \) |
|---|---|---|---|---|
| 1 | 0.566 | 0.699 | 0.673 | 0.644 |
| 2 | 0.352 | 0.448 | 0.459 | 0.415 |
| 3 | 0.172 | 0.163 | 0.169 | 0.168 |
| 4 | 0.042 | 0.005 | 0.021 | 0.022 |
| 5 | 0.285 | 0.755 | 0.745 | 0.816 |
| 6 | 0.230 | 0.434 | 0.406 | 0.367 |
| 7 | 0.268 | 0.269 | 0.307 | 0.292 |

MSD, respectively, when \( L = 2 \) and \( L = 3 \). The estimation errors can be both positive or negative depending on the AP order, the step-size, and the odd or even sample times. On the contrary, for the AP order \( L = 1 \), the estimations are inaccurate. The large estimation errors for \( L = 1 \) are due to the bad conditioning of the matrices \( \mathbf{M}_1 - \mathbf{I} \) that takes to a poor estimate of the asymptotic solution. For larger AP orders, the data reuse property of the AP algorithm takes to more regular matrices \( \mathbf{M}_L \). Indeed, Table 4 compares the condition number, that is, the ratio between the magnitude of the largest and the smallest of the eigenvalues of the matrix \( \mathbf{M}_L - \mathbf{I} \) of the nonlinear controller at even-time indexes for the AP orders \( L = 1, 2, 3 \) and for different values of the step-size.

Figures 5 and 6 diagram the ensemble averages, estimated over 100 runs of the FX-PE-AP and the FX-AP algorithms with step-size equal to 0.032, of the mean value of the residual power of the error computed on 100 successive samples for the nonlinear controller with \( I = 1, J = 2, K = 3 \), and with \( I = 1, J = 2, K = 4 \), respectively. In the case \( I = 1, J = 2, K = 3 \), the transfer functions of the primary paths, \( \mathbf{p}_{1,1}(z) \) and \( \mathbf{p}_{2,1}(z) \), and of the secondary paths, \( \mathbf{s}_{1,1}(z), \mathbf{s}_{1,2}(z), \mathbf{s}_{2,1}(z) \), and \( \mathbf{s}_{2,2}(z) \), are given by (35)–(36), while the other primary and secondary paths are given by

\[
\begin{align*}
\mathbf{p}_{3,1}(z) &= 1.0z^{-2} - 0.3z^{-3} + 0.1z^{-4}, \\
\mathbf{s}_{3,1}(z) &= 1.6z^{-1} - 0.6z^{-2} + 0.1z^{-3}, \\
\mathbf{s}_{3,2}(z) &= 1.6z^{-1} - 0.2z^{-2} - 0.1z^{-3}.
\end{align*}
\]

In the case \( I = 1, J = 2, K = 4 \), the transfer functions of the primary paths, \( \mathbf{p}_{1,1}(z), \mathbf{p}_{2,1}(z), \mathbf{p}_{3,1}(z) \), and of the secondary paths, \( \mathbf{s}_{1,1}(z), \mathbf{s}_{1,2}(z), \mathbf{s}_{2,1}(z), \mathbf{s}_{2,2}(z), \mathbf{s}_{3,1}(z) \), and \( \mathbf{s}_{3,2}(z) \), are given by (35)–(37), and the other primary and secondary paths are given by

\[
\begin{align*}
\mathbf{p}_{4,1}(z) &= 1.0z^{-2} - 0.2z^{-3} + 0.2z^{-4}, \\
\mathbf{s}_{4,1}(z) &= 1.3z^{-1} - 0.5z^{-2} - 0.2z^{-3}, \\
\mathbf{s}_{4,2}(z) &= 1.3z^{-1} - 0.4z^{-2} + 0.2z^{-3}.
\end{align*}
\]

All the other experimental conditions are the same of the case \( I = 1, J = 2, K = 2 \). Figures 7 and 8 confirm again that for \( \mu = 0.032 \), the FX-PE-AP algorithm has a convergence speed that is reduced by a factor \( K \) with respect to the approximate FX-AP algorithm. Nevertheless, we must point out that for larger values of the step-size, the reduction of convergence speed of the FX-PE-AP algorithm can be even larger than a factor \( K \).

We have also performed the same simulations by reducing the SNR at the error microphones to 30, 20, and 10 dB and we have obtained similar convergence behaviors. The main difference, apart from the increase in the residual error, has been that the lowest is the SNR at the error microphones, the lowest is the improvement in the convergence speed obtained by increasing the affine projection order.
Figure 3: Theoretical ( - - ) and simulation values (–) of steady-state MSE versus step-size of the FX-PE-AP algorithm (a) at even samples with a nonlinear controller, (b) at odd samples with a nonlinear controller, (c) at even samples with a linear controller, (d) at odd samples with a linear controller, for \( L = 1, 2, \) and 3.
Figure 4: Theoretical (- -) and simulation values (–) of steady-state MSD versus step-size of the FX-PE-AP algorithm (a) at even samples with a nonlinear controller, (b) at odd samples with a nonlinear controller, (c) at even samples with a linear controller, (d) at odd samples with a linear controller, for $L = 1, 2,$ and $3$. 
Figure 5: Evolution of residual power of the error of (a) the FX-PE-AP algorithm and (b) FX-AP algorithm with a nonlinear controller and $I = 1, J = 2, K = 2$. The dashed lines diagram the asymptotic values of the residual power.

Figure 6: Evolution of residual power of the error of (a) the FX-PE-AP algorithm and (b) FX-AP algorithm with a linear controller and $I = 1, J = 2, K = 2$. The dashed lines diagram the asymptotic values of the residual power.

Figure 7: Evolution of residual power of the error of (a) the FX-PE-AP algorithm and (b) FX-AP algorithm with a nonlinear controller and $I = 1, J = 2, K = 3$. The dashed lines diagram the asymptotic values of the residual power.
6. CONCLUSION

In this paper, we have provided an analysis of the transient and the steady-state behavior of the FX-PE-AP algorithm. We have shown that the algorithm in presence of stationary input signals converges to a cyclostationary process, that is, the asymptotic value of the coefficient vector, the mean-square error and the mean-square deviation tend to periodic functions of the sample time. We have shown that the asymptotic coefficient vector of the FX-PE-AP algorithm differs from the minimum-mean-square solution of the ANC problem and from the asymptotic solution of the AP algorithm from which the FX-PE-AP algorithm was derived. We have proved that the transient behavior of the algorithm can be studied by the cascade of two linear systems. By studying the system matrices of these two linear systems, we can predict the stability and the convergence speed of the algorithm. Expressions have been derived for the steady-state MSE and MSD of the FX-PE-AP algorithm. Eventually, we have compared the FX-PE-AP with the approximate FX-AP algorithm introduced in [4]. Compared with the approximate FX-AP algorithm, the FX-PE-AP algorithm is capable of reducing the adaptation complexity with a factor $K$. Nevertheless, also the convergence speed of the algorithm reduces of the same value.

APPENDIX

PROOF OF THEOREM 1

If we apply the expectation operator to both sides of (23), and if we take into account the hypothesis in (A1), we can derive the result of

$$
E[\mu(mK+i)]^2
=E[\mu(mK+i)]^2 - 2E[\mu(mK+i)]E[qz_i(mK+i)]
+E[\mu_t(mK+i)\Sigma_{ij}(mK+i)],
$$

(A.5)

where

$$
\Sigma_{ij} = E[M_i^T(n)M_j(n)].
$$

(A.2)

Moreover, under the same hypothesis (A1), the evolution of the mean of the coefficient vector from (15) is described by

$$
E[\mu(mK+i)] = E[M_i(mK+i)]E[\mu(mK+i)]
- E[\mu_i(mK+i)].
$$

(A.3)

We manipulate (A.1), (A.2), and (A.3) by taking advantage of the properties of the vector operator $\text{vec}\{\cdot\}$ and of the Kronecker product, $\otimes$. We introduce the vectors $\sigma = \text{vec}\{\Sigma\}$ and $\sigma' = \text{vec}\{\Sigma'\}$. Since for any matrices $A$, $B$, and $C$, it is

$$
\text{vec}\{ABC\} = (C^T \otimes A) \text{vec}\{B\},
$$

(A.4)

we have from (A.2) that

$$
\sigma' = F_i \sigma
$$

(A.5)
where $F_i$ is the $M^2 \times M^2$ matrix defined by

$$F_i = E[M_i^T(n) \otimes M_i^T(n)]. \quad (A.6)$$

The product $E[q_{i,m}(n)]E[w(n)]$ can be evaluated as in

$$E[w^T(n)]E[q_{i,m}(n)] = \text{Tr} \left[ E[w^T(n)]E[q_{i,m}(n)] \right] = E[w^T(n)] \text{vec} \left[ E[q_{i,m}(n)] \right], \quad (A.7)$$

with

$$\text{vec} \left[ E[q_{i,m}(n)] \right] = \text{vec} \left[ E[M_i(n) M_i(n)] \right] = E[M_i(n) \otimes M_i(n)] \sigma = Q_i \sigma, \quad (A.8)$$

and the $M \times M^2$ matrix $Q_i$ is given by

$$Q_i = E[M_i(n) \otimes M_i(n)]. \quad (A.9)$$

Moreover, the last term of (A.1) can be computed as in

$$\text{Tr} \left[ E[m_i^T(n) \Sigma m_i(n)] \right] = g_i^T \sigma, \quad (A.10)$$

where

$$g_i = \text{vec} \left[ E[m_i(n) m_i^T(n)] \right]. \quad (A.11)$$

Accordingly, introducing $\sigma$ and $\sigma'$ instead of $\Sigma$ and $\Sigma'$ and using the results of (A.5), (A.7), (A.8), and (A.10), the recursion in (A.1) can be rewritten as follows:

$$E\left[ \left\| w(mK + i + K) \right\|_{\Sigma^{-1} \{F_i \sigma \}}^2 \right] = E\left[ \left\| w(mK + i) \right\|_{\Sigma^{-1} \{F_i \sigma \}}^2 \right] - 2E[w^T(mK + i)]Q_i \sigma + g_i^T \sigma. \quad (A.12)$$

The recursion in (A.12) shows that in order to evaluate $E\left[ \left\| w(mK + i + K) \right\|_{\Sigma^{-1} \{F_i \sigma \}}^2 \right]$, we need $E\left[ \left\| w(mK + i) \right\|_{\Sigma^{-1} \{F_i \sigma \}}^2 \right]$. This quantity can be inferred from (A.12) by replacing $\sigma$ with $F_i \sigma$, obtaining the following relation:

$$E\left[ \left\| w(mK + i + K) \right\|_{\Sigma^{-1} \{F_i \sigma \}}^2 \right] = E\left[ \left\| w(mK + i) \right\|_{\Sigma^{-1} \{F_i \sigma \}}^2 \right] - 2E[w^T(mK + i)]Q_i F_i \sigma + g_i^T F_i \sigma. \quad (A.13)$$

This procedure is repeated until we obtain the following expression [12, 18, 19]:

$$E\left[ \left\| w(mK + i + K) \right\|_{\Sigma^{-1} \{F_i \sigma \}}^2 \right] = E\left[ \left\| w(mK + i) \right\|_{\Sigma^{-1} \{F_i \sigma \}}^2 \right] - 2E[w^T(mK + i)]Q_i F_i^{M^2-1} \sigma + g_i^T F_i^{M^2-1} \sigma. \quad (A.14)$$

According to the Cayley-Hamilton theorem, the matrix $F_i$ satisfies its own characteristic equation. Therefore, if we indicate with $p_i(x)$ the characteristic polynomial of $F_i$, $p_i(x) = \det(xI - F_i)$, for the Cayley–Hamilton theorem we have that $p_i(F_i) = 0$. The characteristic polynomial $p_i(x)$ is an order $M^2$ polynomial that can be written as in

$$p_i(x) = x^{M^2} + p_{M^2-1} x^{M^2-1} + \cdots + p_{0,i}, \quad (A.15)$$

where we indicate with $\{p_{j,i}\}$ the coefficients of the polynomial. Since $p_i(F_i) = 0$, we deduce that [12, 18, 19]

$$E\left[ \left\| w(n) \right\|_{\Sigma^{-1} \{F_i \sigma \}}^2 \right] = - \sum_{j=0}^{M^2-1} p_{j,i} E\left[ \left\| w(n) \right\|_{\Sigma^{-1} \{F_i \sigma \}}^2 \right]. \quad (A.16)$$

The results of (A.3), (A.12)–(A.14), and (A.16) prove Theorem 1 that describes the transient behavior of the FX-PE-AP algorithms.

**ACKNOWLEDGMENT**

This work was supported by MIUR under Grant PRIN 2004092314.

**REFERENCES**

[1] P. A. Nelson and S. J. Elliott, *Active Control of Sound*, Academic Press, London, UK, 1995.

[2] S. C. Douglas, “Fast implementations of the filtered-X LMS and LMS algorithms for multichannel active noise control,” *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 4, pp. 454–465, 1999.

[3] M. Bouchard, “Multichannel affine and fast affine projection algorithms for active noise control and acoustic equalization systems,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 1, pp. 54–60, 2003.

[4] A. Carini and G. L. Sicuranza, “Transient and steady-state analysis of filtered-x affine projection algorithms,” *IEEE Transactions on Signal Processing*, vol. 54, no. 2, pp. 665–678, 2006.

[5] Y. Neuvo, C.-Y. Dong, and S. K. Mitra, “Interpolated finite impulse response filters,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 3, pp. 563–570, 1984.

[6] S. Werner and P. S. R. Diniz, “Set-membership affine projection algorithm,” *IEEE Signal Processing Letters*, vol. 8, no. 8, pp. 231–235, 2001.

[7] S. C. Douglas, “Adaptive filters employing partial updates,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 3, pp. 209–216, 1997.

[8] K. Doğançay and O. Tanrikulu, “Adaptive filtering algorithms with selective partial updates,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 8, pp. 762–769, 2001.

[9] G. L. Sicuranza and A. Carini, “Nonlinear multichannel active noise control using partial updates,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’05)*, vol. 3, pp. 109–112, Philadelphia, PA, USA, March 2005.

[10] E. Björnason, “Analysis of the filtered-X LMS algorithm,” *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 6, pp. 504–514, 1995.

[11] O. J. Tobias, J. C. M. Bermudez, and N. J. Bershad, “Mean weight behavior of the filtered-X LMS algorithm,” *IEEE Transactions on Signal Processing*, vol. 48, no. 4, pp. 1061–1075, 2000.
[12] H.-C. Shin and A. H. Sayed, “Mean-square performance of a family of affine projection algorithms,” IEEE Transactions on Signal Processing, vol. 52, no. 1, pp. 90–102, 2004.

[13] M. Bouchard and S. Quednau, “Multichannel recursive-least-squares algorithms and fast-transversal-filter algorithms for active noise control and sound reproduction systems,” IEEE Transactions on Speech and Audio Processing, vol. 8, no. 5, pp. 606–618, 2000.

[14] V. J. Mathews and G. L. Sicuranza, Polynomial Signal Processing, John Wiley & Sons, New York, NY, USA, 2000.

[15] P. Strauch and B. Mulgrew, “Active control of nonlinear noise processes in a linear duct,” IEEE Transactions on Signal Processing, vol. 46, no. 9, pp. 2404–2412, 1998.

[16] D. P. Das and G. Panda, “Active mitigation of nonlinear noise processes using a novel filtered-s LMS algorithm,” IEEE Transactions on Signal Processing, vol. 49, no. 8, pp. 1667–1676, 2001.
Research Article

Step Size Bound of the Sequential Partial Update LMS Algorithm with Periodic Input Signals

Pedro Ramos, Roberto Torrubia, Ana López, Ana Salinas, and Enrique Masgrau

1 Communication Technologies Group, Aragón Institute for Engineering Research (I3A), EUPT, University of Zaragoza, Ciudad Escolar s/n, 44003 Teruel, Spain
2 Communication Technologies Group, Aragón Institute for Engineering Research (I3A), CPS Ada Byron, University of Zaragoza, Maria de Luna 1, 50018 Zaragoza, Spain

Received 9 June 2006; Revised 2 October 2006; Accepted 5 October 2006

Recommended by Kutluyıl Dogancay

This paper derives an upper bound for the step size of the sequential partial update (PU) LMS adaptive algorithm when the input signal is a periodic reference consisting of several harmonics. The maximum step size is expressed in terms of the gain in step size of the PU algorithm, defined as the ratio between the upper bounds that ensure convergence in the following two cases: firstly, when only a subset of the weights of the filter is updated during every iteration; and secondly, when the whole filter is updated at every cycle. Thus, this gain in step-size determines the factor by which the step size parameter can be increased in order to compensate the inherently slower convergence rate of the sequential PU adaptive algorithm. The theoretical analysis of the strategy developed in this paper excludes the use of certain frequencies corresponding to notches that appear in the gain in step size. This strategy has been successfully applied in the active control of periodic disturbances consisting of several harmonics, so as to reduce the computational complexity of the control system without either slowing down the convergence rate or increasing the residual error. Simulated and experimental results confirm the expected behavior.

Copyright © 2007 Pedro Ramos et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

1.1. Context of application: active noise control systems

Acoustic noise reduction can be achieved by two different methods. Passive techniques are based on the absorption and reflection properties of materials, showing excellent noise attenuation for frequencies above 1 kHz. Nevertheless, passive sound absorbers do not work well at low frequencies because the acoustic wavelength becomes large compared to the thickness of a typical noise barrier. On the other hand, active noise control (ANC) techniques are based on the principle of destructive wave interference, whereby an antinoise is generated with the same amplitude as the undesired disturbance but with an appropriate phase shift in order to cancel the primary noise at a given location, generating a zone of silence around an acoustical sensor.

The basic idea behind active control was patented by Lueg [1]. However, it was with the relatively recent advent of powerful and inexpensive digital signal processors (DSPs) that ANC techniques became practical because of their capacity to perform the computational tasks involved in real time.

The most popular adaptive algorithm used in DSP-based implementations of ANC systems is the filtered-x least mean-square (FxLMS) algorithm, originally proposed by Morgan [2] and independently derived by Widrow et al. [3] in the context of adaptive feedforward control and by Burgess [4] for the active control of sound in ducts. Figure 1 shows the arrangement of electroacoustic elements and the block diagram of this well known solution, aimed at attenuating acoustic noise by means of secondary sources. Due to the presence of a secondary path transfer function following the adaptive filter, the conventional LMS algorithm must be modified to ensure convergence. The mentioned secondary path includes the D/A converter, power amplifier, loudspeaker, acoustic path, error microphone, and A/D converter. The solution proposed by the FxLMS is based on the placement of an accurate estimate of the secondary path transfer function in the weight update path as originally suggested in [2]. Thus, the regressor signal of the adaptive filter...
is obtained by filtering the reference signal through the estimate of the secondary path.

### 1.2. Partial update LMS algorithm

The LMS algorithm and its filtered-x version have been widely used in control applications because of their simple implementation and good performance. However, the adaptive FIR filter may eventually require a large number of coefficients to meet the requirements imposed by the addressed problem. For instance, in the ANC system described in Figure 1(b), the task associated with the adaptive filter—in order to minimize the error signal—is to accurately model the primary path and inversely model the secondary path. Previous research in the field has shown that if the active canceller has to deal with an acoustic disturbance consisting of closely spaced frequency harmonics, a long adaptive filter is necessary [5]. Thus, an improvement in performance is achieved at the expense of increasing the computational load of the control strategy. Because of limitations in computational efficiency and memory capacity of low-cost DSP boards, a large number of coefficients may even impair the practical implementation of the LMS or more complex adaptive algorithms.

As an alternative to the reduction of the number of coefficients, one may choose to update only a portion of the filter coefficient vector at each sample time. Partial update (PU) adaptive algorithms have been proposed to reduce the large computational complexity associated with long adaptive filters. As far as the drawbacks of PU algorithms are concerned, it should be noted that their convergence speed is reduced approximately in proportion to the filter length divided by the number of coefficients updated per iteration, that is, the decimation factor \( N \). Therefore, the tradeoff between convergence performance and complexity is clearly established: the larger the saving in computational costs, the slower the convergence rate.

Two well-known adaptive algorithms carry out the partial updating process of the filter vector employing decimated versions of the error or the regressor signals [6]. These algorithms are, respectively, the periodic LMS and the sequential LMS. This work focuses the attention on the later.

The sequential LMS algorithm with decimation factor \( N \) updates a subset of size \( L/N \), out of a total of \( L \), coefficients per iteration according to (1),

\[
w_l(n + 1) = \begin{cases} 
  w_l(n) + \mu x(n - l + 1)e(n) & \text{if } (n - l + 1) \mod N = 0, \\
  w_l(n) & \text{otherwise} 
\end{cases}
\]

for \( 1 \leq l \leq L \), where \( w_l(n) \) represents the \( l \)th weight of the filter, \( \mu \) is the step size of the adaptive algorithm, \( x(n) \) is the regressor signal, and \( e(n) \) is the error signal.

The reduction in computational costs of the sequential PU strategy depends directly on the decimation factor \( N \).
Tables 1 and 2 show, respectively, the computational complexity of the LMS and the sequential LMS algorithms in terms of the average number of operations required per cycle, when used in the context of a filtered-x implementation of a single-channel ANC system. The length of the adaptive filter is $L$, the length of the offline estimate of the secondary path is $L_0$, and the decimation factor is $N$.

The criterion for the selection of coefficients to be updated can be modified and, as a result of that, different PU adaptive algorithms have been proposed [7–10]. The variations of the cited PU LMS algorithms speed up their convergence rate at the expense of increasing the number of operations per cycle. These extra operations include the “intelligence” required to optimize the election of the coefficients to be updated at every instant.

In this paper, we try to go a step further, showing that in applications based on the sequential LMS algorithm, where the regressor signal is periodic, the inclusion of a new parameter—called gain in step size—in the traditional trade-off proves that one can achieve a significant reduction in the computational costs without degrading the performance of the algorithm. The proposed strategy—filtered-x sequential least mean-square algorithm with gain in step size ($G_x$-FxLMS)—has been successfully applied in our laboratory in the context of active control of periodic noise [5].

### 1.3. Assumptions in the convergence analysis

Before focusing on the sequential PU LMS strategy and the derivation of the gain in step size, it is necessary to remark on two assumptions about the upcoming analysis: the independence theory and the slow convergence condition.

The traditional approach to convergence analyses of LMS—and FxLMS—algorithms is based on stochastic inputs instead of deterministic signals such as a combination of multiple sinusoids. Those stochastic analyses assume independence between the reference—or regressor—signal and the coefficients of the filter vector. In spite of the fact that this independence assumption is not satisfied or, at least, questionable when the reference signal is deterministic, some researchers have previously used the independence assumption with a deterministic reference. For instance, Kuo et al. [11] assumed the independence theory, the slow convergence condition, and the exact offline estimate of the secondary path to state that the maximum step size of the FxLMS algorithm is inversely bounded by the maximum eigenvalue of the autocorrelation matrix of the filtered reference, when the reference was considered to be the sum of multiple sinusoids. Bjarnason [12] used as well the independence theory to carry out a FxLMS analysis extended to a sinusoidal input. According to Bjarnason, this approach is justified by the fact that experience with the LMS algorithm shows that results obtained by the application of the independence theory retain sufficient information about the structure of the adaptive process to serve as reliable design guidelines, even for highly dependent data samples.

As far as the second assumption is concerned, in the context of the traditional convergence analysis of the FxLMS adaptive algorithm [13, Chapter 3], it is necessary to assume slow convergence—i.e., that the control filter is changing slowly—and to count on an exact estimate of the secondary path in order to commute the order of the adaptive filter and the secondary path [2]. In so doing, the output of the adaptive filter carries through directly to the error signal, and the traditional LMS algorithm analysis can be applied by using as regressor signal the result of the filtering of the reference signal through the secondary path transfer function.

It could be argued that this condition compromises the determination of an upper bound on the step size of the adaptive algorithm, but actually, slow convergence is guaranteed because the convergence factor is affected by a much more restrictive condition with a periodic reference than with a white noise reference. It has been proved that with a sinusoidal reference, the upper bound of the step size is inversely proportional to the product of the length of the filter and the delay in the secondary path; whereas with a white reference signal, the bound depends inversely on the sum of these parameters, instead of their product [12, 14].

Simulations with a white noise reference signal suggest that a realistic upper bound in the step size is given by [15, Chapter 3]

$$
\mu_{\text{max}} = \frac{2}{P_X(L + \Delta)},
$$

where $P_X$ is the power of the filtered reference, $L$ is the length of the adaptive filter, and $\Delta$ is the delay introduced by the secondary path.

Bjarnason [12] analyzed FxLMS convergence with a sinusoidal reference, but employed the habitual assumptions made with stochastic signals, that is, the independence theory. The stability condition derived by Bjarnason yields

$$
\mu_{\text{max}} = \frac{2}{P_X L} \sin \left( \frac{\pi}{2(2\Delta + 1)} \right). \quad (4)
$$

In case of large delay $\Delta$, (3) simplifies to

$$
\mu_{\text{max}} \approx \frac{\pi}{P_X L(2\Delta + 1)}, \quad \Delta \gg \frac{\pi}{4}. \quad (4)
$$

Vicente and Masgrau [14] obtained an upper bound for the FxLMS step size that ensures convergence when the reference signal is deterministic (extended to any combination of multiple sinusoids). In the derivation of that result, there is no need of any of the usual approximations, such as independence between reference and weights or slow convergence. The maximum step size for a sinusoidal reference is given by

$$
\mu_{\text{max}} = \frac{2}{P_X L(2\Delta + 1)}. \quad (5)
$$

The similarity between both convergence conditions—(4) and (5)—is evident in spite of the fact that the former analysis is based on the independence assumption, whereas the latter analysis is exact. This similarity achieved in the results justifies the use of the independence theory when dealing with sinusoidal references, just to obtain a first-approach
Many convergence analyses of the LMS algorithm try to derive exact bounds on the step size to guarantee mean and mean-square convergence based on the independence assumption [16, Chapter 6]. Analyses based on such assumption have been extended to sequential PU algorithms [6] to yield the following result: the bounds on the step size for the sequential LMS algorithm are the same as those for the LMS algorithm and, as a result of that, a larger step size cannot be used in order to compensate its inherently slower convergence rate. However, this result is only valid for independent identically distributed (i.i.d.) zero-mean Gaussian input signals.

To obtain a valid analysis in the case of periodic signals as input of the adaptive filter, we will focus on the updating process of the coefficients when the $L$-length filter is adapted by the sequential LMS algorithm with decimation factor $N$. This algorithm updates just $L/N$ coefficients per iteration according to (1). For ease in analyzing the PU strategy, it is assumed throughout the paper that $L/N$ is an integer.

Figure 1(b) shows the block diagram of a filtered-x ANC system, where the secondary path $S(z)$ is placed following the digital filter $W(z)$ controlled by an adaptive algorithm. As has been previously stated, under the assumption of slow convergence and considering an accurate offline estimate of the secondary path, the order of $W(z)$ and $S(z)$ can be commuted and the resulting equivalent diagram simplified. Thus, standard LMS algorithm techniques can be applied to the filtered-x version of the sequential LMS algorithm in order to determine the convergence of the mean weights and the maximum value of the step size [13, Chapter 3]. The simplified analysis is based on the consideration of the filtered reference as the regressor signal of the adaptive filter. This signal is denoted as $x'(n)$ in Figure 1(b).

Figure 2 summarizes the sequential PU algorithm given by (1), indicating the coefficients to be updated at each iteration and the related samples of the regressor signal used in each update, $x'(n)$ being the value of the regressor signal at the current instant.

**2. EIGENVALUE ANALYSIS OF PERIODIC NOISE: THE GAIN IN STEP SIZE**

**2.1. Overview**

Many convergence analyses of the LMS algorithm try to derive exact bounds on the step size to guarantee mean and mean-square convergence based on the independence assumption [16, Chapter 6]. Analyses based on such assumption have been extended to sequential PU algorithms [6] to yield the following result: the bounds on the step size for the sequential LMS algorithm are the same as those for the LMS algorithm and, as a result of that, a larger step size cannot be used in order to compensate its inherently slower convergence rate. However, this result is only valid for independent identically distributed (i.i.d.) zero-mean Gaussian input signals.

To obtain a valid analysis in the case of periodic signals as input of the adaptive filter, we will focus on the updating process of the coefficients when the $L$-length filter is adapted by the sequential LMS algorithm with decimation factor $N$. This algorithm updates just $L/N$ coefficients per iteration according to (1). For ease in analyzing the PU strategy, it is assumed throughout the paper that $L/N$ is an integer.

Figure 1(b) shows the block diagram of a filtered-x ANC system, where the secondary path $S(z)$ is placed following the digital filter $W(z)$ controlled by an adaptive algorithm. As has been previously stated, under the assumption of slow convergence and considering an accurate offline estimate of the secondary path, the order of $W(z)$ and $S(z)$ can be commuted and the resulting equivalent diagram simplified. Thus, standard LMS algorithm techniques can be applied to the filtered-x version of the sequential LMS algorithm in order to determine the convergence of the mean weights and the maximum value of the step size [13, Chapter 3]. The simplified analysis is based on the consideration of the filtered reference as the regressor signal of the adaptive filter. This signal is denoted as $x'(n)$ in Figure 1(b).

Figure 2 summarizes the sequential PU algorithm given by (1), indicating the coefficients to be updated at each iteration and the related samples of the regressor signal used in each update, $x'(n)$ being the value of the regressor signal at the current instant.
the filtered reference signal. However, according to Figure 2, the sequential LMS algorithm uses only every Nth element of the regressor signal. Thus, it is not worth computing a new sample of the filtered reference at every algorithm iteration. It is enough to obtain the value of a new sample at just one out of N iterations.

The L-length filter can be considered as formed by N sub-filters of L/N coefficients each. These sub-filters are obtained by uniformly sampling by N the weights of the original vector. Coefficients of the first subfilter are encircled in Figure 2. Hence, the whole updating process can be understood as the N-cyclical updating schedule of N subfilters of length L/N. Coefficients occupying the same relative position in every subfilter are updated with the same sample of the regressor signal. This regressor signal is only renewed at one in every N iterations. That is, after N iterations, the less recent value is shifted out of the valid range and a new value is acquired and subsequently used to update the first coefficient of each subfilter.

To sum up, during N consecutive instants, N subfilters of length L/N are updated with the same regressor signal. This regressor signal is a N-decimated version of the filtered reference signal. Therefore, the overall convergence can be analyzed on the basis of the joint convergence of N subfilters:

(i) each of length L/N,
(ii) updated by an N-decimated regressor signal.

2.2. Spectral norm of autocorrelation matrices: the triangle inequality

The autocorrelation matrix \( R \) of a periodic signal consisting of several harmonics is Hermitian and Toeplitz.

The spectral norm of a matrix \( A \) is defined as the square root of the largest eigenvalue of the matrix product \( A^H A \), where \( A^H \) is the Hermitian transpose of \( A \), that is, [17, Appendix E]

\[
\| A \|_s = \left[ \lambda_{\text{max}} (A^H A) \right]^{1/2}. \tag{6}
\]

The spectral norm of a matrix satisfies, among other norm conditions, the triangle inequality given by

\[
\| A + B \|_s \leq \| A \|_s + \| B \|_s. \tag{7}
\]

The application of the definition of the spectral norm to the Hermitian correlation matrix \( R \) leads us to conclude that

\[
\| R \|_s = \left[ \lambda_{\text{max}} (R^H R) \right]^{1/2} = \left[ \lambda_{\text{max}} (R R^H) \right]^{1/2} = \lambda_{\text{max}} (R). \tag{8}
\]

Therefore, since \( A \) and \( B \) are correlation matrices, we have the following result:

\[
\lambda_{\text{max}} (A + B) = \| A + B \|_s \leq \| A \|_s + \| B \|_s = \lambda_{\text{max}} (A) + \lambda_{\text{max}} (B). \tag{9}
\]

2.3. Gain in step size for periodic input signals

At this point, a convergence analysis is carried out in order to derive a bound on the step size of the filtered-x sequential PU LMS algorithm when the regressor vector is a periodic signal consisting of multiple sinusoids.

It is known that the LMS adaptive algorithm converges in mean to the solution if the step size satisfies [16, Chapter 6]

\[
0 < \mu < \frac{2}{\lambda_{\text{max}}}, \tag{10}
\]

where \( \lambda_{\text{max}} \) is the largest eigenvalue of the input autocorrelation matrix

\[
R = E[x^T(n)x(n)], \tag{11}
\]

\( x(n) \) being the regressor signal of the adaptive algorithm.

As has been previously stated, under the assumptions considered in Section 1.3, in the case of an ANC system based on the FxLMS, traditional LMS algorithm analysis can be used considering that the regressor vector corresponds to the reference signal filtered by an estimate of the secondary path. The proposed analysis is based on the ratio between the largest eigenvalue of the autocorrelation matrix of the regressor signal for two different situations. Firstly, when the adaptive algorithm is the full update LMS and, secondly, when the updating strategy is based on the sequential LMS algorithm with a decimation factor \( N > 1 \). The sequential LMS with \( N = 1 \) corresponds to the LMS algorithm.

Let the regressor vector \( x(n) \) be formed by a periodic signal consisting of \( K \) harmonics of the fundamental frequency \( f_0 \),

\[
x(n) = \sum_{k=1}^{K} C_k \cos (2\pi k f_0 n + \phi_k). \tag{12}
\]

The autocorrelation matrix of the whole signal can be expressed as the sum of \( K \) simpler matrices with each being the autocorrelation matrix of a single tone [11]

\[
R = \sum_{k=1}^{K} \sum_{k=1}^{K} R^2_k, \tag{13}
\]

where

\[
R^2_k = \begin{bmatrix}
1 & \cos (2\pi k f_0) & \cdots & \cos [2\pi k (L-1) f_0] \\
\cos (2\pi k f_0) & 1 & \cdots & \cos [2\pi k (L-2) f_0] \\
\vdots & \vdots & \ddots & \vdots \\
\cos [2\pi k (L-1) f_0] & \cdots & \cdots & 1
\end{bmatrix}. \tag{14}
\]

If the simple LMS algorithm is employed, the largest eigenvalue of each simple matrix \( R_k \) is given by [11]

\[
\lambda_{k_{\text{max}}} = \max \left\{ \frac{1}{4} \left[ L \pm \frac{\sin (2\pi k f_0)}{\sin (2\pi k f_0)} \right] \right\}. \tag{15}
\]

According to (9) the largest eigenvalue of a sum of matrices is bounded by the sum of the largest eigenvalues of each of
its components. Therefore, the largest eigenvalue of \( R \) can be expressed as

\[
\lambda_{\text{tot}, \text{max}}^{N=1} \leq \sum_{k=1}^{K} C_k^2 \lambda_{k, \text{max}}^{N=1} (k) = \sum_{k=1}^{K} C_k^2 \max \left\{ \frac{1}{4} \left[ \frac{L}{N} \pm \frac{\sin ((L/N)2\pi k f_0)}{\sin (2\pi k f_0)} \right] \right\}. \tag{16}
\]

At the end of Section 2.1, two key differences were derived in the case of the sequential LMS algorithm: the convergence condition of the whole filter might be translated to the parallel convergence of \( N \) subfilters of length \( L/N \) adapted by an \( N \)-decimated regressor signal. Considering both changes, the largest eigenvalue of each simple matrix \( R_k \) can be expressed as

\[
\lambda_{k, \text{max}}^{N=1} (k) = \max \left\{ \frac{1}{4} \left[ \frac{L}{N} \pm \frac{\sin ((L/N)2\pi k f_0)}{\sin (2\pi k f_0)} \right] \right\}. \tag{17}
\]

and considering the triangle inequality (9), we have

\[
\lambda_{\text{tot}, \text{max}}^{N>1} \leq \sum_{k=1}^{K} C_k^2 \lambda_{k, \text{max}}^{N>1} (k) = \sum_{k=1}^{K} C_k^2 \max \left\{ \frac{1}{4} \left[ \frac{L}{N} \pm \frac{\sin ((L/N)2\pi k f_0)}{\sin (2\pi k f_0)} \right] \right\}. \tag{18}
\]

Defining the gain in step size \( G_\mu \) as the ratio between the bounds on the step sizes in both cases, we obtain the factor by which the step size parameter can be multiplied when the adaptive algorithm uses PU,

\[
G_\mu (K, f_0, L, N) = \frac{\mu_{N>1}^{\text{tot}, \text{max}}}{\mu_{N=1}^{\text{tot}, \text{max}}} = \frac{2}{\max \{ \lambda_{N=1}^{\text{tot}, \text{max}} \}} = \frac{\sum_{k=1}^{K} C_k^2 \lambda_{k, \text{max}}^{N=1} (k)}{\sum_{k=1}^{K} C_k^2 \lambda_{k, \text{max}}^{N>1} (k)} = \frac{\sum_{k=1}^{K} C_k^2 \max \{ (1/4) \left[ L \pm \frac{\sin ((L/N)2\pi k f_0)}{\sin (2\pi k f_0)} \right] \}}{\sum_{k=1}^{K} C_k^2 \max \{ (1/4) \left[ L/N \pm \frac{\sin ((L/N)2\pi k f_0)}{\sin (2\pi k f_0)} \right] \}}. \tag{19}
\]

In order to more easily visualize the dependence of the gain in step size on the length of the filter \( L \) and on the decimation factor \( N \), let a single tone of normalized frequency \( f_0 \) be the regressor signal

\[
x' (n) = \cos (2\pi f_0 n + \phi). \tag{20}
\]

Now, the gain in step size, that is, the ratio between the bounds on the step size when \( N > 1 \) and \( N = 1 \), is given by

\[
G_\mu (1, f_0, L, N) = \frac{\mu_{N>1}^{\text{max}}}{\mu_{N=1}^{\text{max}}} = \frac{\max \{ (1/4) \left[ L \pm \frac{\sin (L_2\pi f_0)}{\sin (2\pi f_0)} \right] \}}{\max \{ (1/4) \left[ L/N \pm \frac{\sin ((L/N)2\pi f_0)}{\sin (2\pi f_0)} \right] \}}. \tag{21}
\]

Figures 3 and 4 show the gain in step size expressed by (21) for different decimation factors \( (N) \) and different lengths of the adaptive filter \( (L) \).

Basically, the analytical expressions and figures show that the step size can be multiplied by \( N \) as long as certain frequencies, at which a notch in the gain in step size appears, are avoided. The location of these critical frequencies, as well as the number and width of the notches, will be analyzed as a function of the sampling frequency \( F_s \), the length of the adaptive filter \( L \), and the decimation factor \( N \). According to (19) and (21), with increasing decimation factor \( N \), the step size can be multiplied by \( N \) and, as a result of that affordable compensation, the PU sequential algorithm convergence is as fast as the full update FxLMS algorithm as long as the undesired disturbance is free of components located at the notches of the gain in step size.

Figure 3 shows that the total number of equidistant notches appearing in the gain in step size is \( (N - 1) \). In fact, the notches appear at the frequencies given by

\[
f_{k, \text{notch}} = k \frac{f_s}{2N}, \quad k = 1, \ldots, N - 1. \tag{22}
\]

It is important to avoid the undesired sinusoidal noise being at the mentioned notches because the gain in step size is smaller there, with the subsequent reduction in convergence rate. As far as the width of the notches is concerned, Figure 4 (where the decimation factor \( N = 2 \)) shows that the smaller the length of the filter, the wider the main notch of the gain in step size. In fact, if \( L/N \) is an integer, the width between first zeros of the main notch can be expressed as

\[
width = \frac{F_s}{L}. \tag{23}
\]

Simulations and practical experiments confirm that at these problematic frequencies, the gain in step size cannot be applied at its maximum value \( N \).

If it were not possible to avoid the presence of some harmonic at a frequency where there were a notch in the gain, the proposed strategy could be combined with the filtered-error least mean-square (FxLMS) algorithm [13, Chapter 3]. The FeLMS algorithm is based on a shaping filter \( C(z) \) placed in the error path and in the filtered reference path. The transfer function \( C(z) \) is the inverse of the desired shape of the residual noise. Therefore, \( C(z) \) must be designed as a comb filter with notches at the problematic frequencies. As a result of that, the harmonics at those frequencies would not be canceled. Nevertheless, if a noise component were to fall in a notch, using a smaller step size could be preferable to using the FeLMS, considering that typically it is more important to cancel all noise disturbance frequencies rather than obtaining the fastest possible convergence rate.

### 3. NOISE ON THE WEIGHT VECTOR SOLUTION AND EXCESS MEAN-SQUARE ERROR

The aim of this section is to prove that the full-strength gain in step size \( G_\mu = N \) can be applied in the context of ANC
systems controlled by the filtered-x sequential LMS algorithm without an additional increase in mean-square error caused by the noise on the weight vector solution. We begin with an analysis of the trace of the autocorrelation matrix of an $N$-decimated signal $x_{N}(n)$, which is included to provide mathematical support for subsequent parts. The second part of the section revises the analysis performed by Widrow and Stearns of the effect of the gradient noise on the LMS algorithm [16, Chapter 6]. The section ends with the extension to the $G_{\mu}$-FxSLMS algorithm of the previously outlined analysis.

### 3.1. Properties of the trace of an $N$-decimated autocorrelation matrix

Let the $L \times 1$ vector $x(n)$ represent the elements of a signal. To show the composition of the vector $x(n)$, we write

$$x(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^T.$$  \hfill (24)

The expectation of the outer product of the vector $x(n)$ with itself determines the $L \times L$ autocorrelation matrix $R$ of the
signal
\[ R = E[x(n)x^T(n)] \]
\[
\begin{bmatrix}
  r_{xx}(0) & r_{xx}(1) & r_{xx}(2) & \cdots & r_{xx}(L-1) \\
  r_{xx}(1) & r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(L-2) \\
  r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(L-3) \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  r_{xx}(L-1) & r_{xx}(L-2) & r_{xx}(L-3) & \cdots & r_{xx}(0)
\end{bmatrix}
\]  
(25)

The \( N \)-decimated signal \( x_k(n) \) is obtained from vector \( x(n) \) by multiplying \( x(n) \) by the auxiliary matrix \( I_k^{(N)} \),
\[ x_k(n) = I_k^{(N)} x(n), \quad k = 1 + n \text{ mod } N, \]  
(26)
where \( I_k^{(N)} \) is obtained from the identity matrix \( I \) of dimension \( L \times L \) by zeroing out some elements in \( I \). The first nonnull element on its main diagonal appears at the \( k \)th position and the superscript \( (N) \) is intended to denote the fact that two consecutive nonzero elements on the main diagonal are separated by \( N \) positions. The auxiliary matrix \( I_k^{(N)} \) is explicitly expressed as
\[
I_k^{(N)} = \begin{pmatrix}
0 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & 0
\end{pmatrix}
\]  
(27)
As a result of (26), the autocorrelation matrix \( R_N \) of the new signal \( x_k(n) \) only presents nonnull elements on its main diagonal and on any other diagonal parallel to the main diagonal that is separated from it by \( kN \) positions, \( k \) being any integer. Thus,
\[ R_N = E[x_N(n)x_N^T(n)] \]
\[
\frac{1}{N} \begin{bmatrix}
  r_{xx}(0) & 0 & \cdots & 0 & r_{xx}(N) & \cdots & r_{xx}(2N) & \cdots \\
  0 & r_{xx}(0) & 0 & \cdots & 0 & r_{xx}(N) & \cdots & \vdots \\
  \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
  0 & \cdots & 0 & r_{xx}(0) & 0 & \cdots & 0 & r_{xx}(N) \\
  r_{xx}(N) & 0 & \cdots & 0 & r_{xx}(0) & 0 & \cdots & 0 \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  r_{xx}(2N) & 0 & \cdots & 0 & r_{xx}(0) & 0 & \cdots & 0 \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{bmatrix}
\]  
(28)

The matrix \( R_N \) can be expressed in terms of \( R \) as
\[ R_N = \frac{1}{N} \sum_{i=1}^{N} I_i^{(N)} R I_i^{(N)}. \]  
(29)

We define the diagonal matrix \( \Lambda \) with main diagonal comprised of the \( L \) eigenvalues of \( R \). If \( Q \) is a matrix whose columns are the eigenvectors of \( R \), we have
\[ \Lambda = Q^{-1} R Q = \begin{pmatrix}
\lambda_1 & 0 & \cdots & 0 \\
0 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & \lambda_L
\end{pmatrix}. \]  
(30)

The trace of \( R \) is defined as the sum of its diagonal elements. The trace can also be obtained from the sum of its eigenvalues, that is,
\[ \text{trace}(R) = \sum_{i=1}^{L} r_{xx}(0) = \text{trace}(\Lambda) = \sum_{i=1}^{L} \lambda_i. \]  
(31)

The relation between the traces of \( R \) and \( R_N \) is given by
\[ \text{trace}(R_N) = \frac{1}{N} \sum_{i=1}^{L} r_{xx}(0) N = \frac{\text{trace}(R)}{N}. \]  
(32)

### 3.2. Effects of the gradient noise on the LMS algorithm

Let the vector \( w(n) \) represent the weights of the adaptive filter, which are updated according to the LMS algorithm as follows:
\[ w(n + 1) = w(n) - \mu \hat{v}(n) = w(n) + \mu e(n) x(n), \]  
(33)
where \( \mu \) is the step size, \( \hat{v}(n) \) is the gradient estimate at the \( n \)th iteration, \( e(n) \) is the error at the previous iteration, and \( x(n) \) is the vector of input samples, also called the regressor signal.

We define \( v(n) \) as the deviation of the weight vector from its optimum value
\[ v(n) = w(n) - w_{\text{opt}}, \]  
(34)
and \( v'(n) \) as the rotation of \( v(n) \) by means of the eigenvector matrix \( Q \),
\[ v'(n) = Q^{-1} v(n) = Q^{-1} [w(n) - w_{\text{opt}}]. \]  
(35)

In order to give a measure of the difference between actual and optimal performance of an adaptive algorithm, two parameters can be taken into account: excess mean-square error and misadjustment. The excess mean-square error \( \xi_{\text{excess}} \) is the average mean-square error less the minimum mean-square error, that is,
\[ \xi_{\text{excess}} = E[\xi(n)] - \xi_{\text{min}}. \]  
(36)
The misadjustment $M$ is defined as the excess mean-square error divided by the minimum mean-square error

$$M = \frac{\xi_{\text{excess}}}{\xi_{\text{min}}} = \frac{E[\xi(n)] - \xi_{\text{min}}}{\xi_{\text{min}}}.$$  (37)

Random weight variations around the optimum value of the filter cause an increase in mean-square error. The average of these increases is the excess mean-square error. Widrow and Stearns [16, Chapters 5 and 6] analyzed the steady-state effects of gradient noise on the weight vector solution of the LMS algorithm by means of the definition of a vector of noise $\mathbf{n}(n)$ in the gradient estimate at the $n$th iteration. It is assumed that the LMS process has converged to a steady-state weight vector solution near its optimum and that the true gradient $\nabla(n)$ is close to zero. Thus, we write

$$\mathbf{n}(n) = \hat{\nabla}(n) - \nabla(n) = \hat{\nabla}(n) = -2e(n)x(n).$$  (38)

The weight vector covariance in the principal axis coordinate system, that is, in primed coordinates, is related to the covariance of the noise as follows [16, Chapter 6]:

$$\text{cov}[v'(n)] = \frac{\mu}{8} \left( A - \frac{\mu}{2} A^2 \right)^{-1} \text{cov}[n'(n)]$$

$$= \frac{\mu}{8} \left( A - \frac{\mu}{2} A^2 \right)^{-1} \text{cov}[Q^{-1}n(n)]$$

$$= \frac{\mu}{8} \left( A - \frac{\mu}{2} A^2 \right)^{-1} Q^{-1}E[n(n)n^T(n)]Q.$$  (39)

In practical situations, $(\mu/2)A$ tends to be negligible with respect to $I$, so that (39) simplifies to

$$\text{cov}[v'(n)] \approx \frac{\mu}{8} A^{-1} Q^{-1} E[n(n)n^T(n)]Q.$$  (40)

From (38), it can be shown that the covariance of the gradient estimation noise of the LMS algorithm at the minimum point is related to the autocorrelation input matrix according to (41)

$$\text{cov}[n(n)] = E[n(n)n^T(n)] = 4E[e^2(n)]R.$$  (41)

In (41), the error and the input vector are considered statistically independent because at the minimum point of the error surface both signals are orthogonal.

To sum up, (40) and (41) indicate that the measurement of how close the LMS algorithm is to optimality in the mean-square error sense depends on the product of the step size and the autocorrelation matrix of the regressor signal $x(n)$.

### 3.3. Effects of gradient noise on the filtered-x sequential LMS algorithm

At this point, the goal is to carry out an analysis of the effect of gradient noise on the weight vector solution for the case of the $G_p$-FxSLMS algorithm in a similar manner as in the previous section.

The weights of the adaptive filter when the $G_p$-FxSLMS algorithm is used are updated according to the recursion

$$\mathbf{w}(n+1) = \mathbf{w}(n) + G_p\mu e(n)\mathbf{I}_{1+n \mod N}^{(N)}x'(n),$$  (42)

where $\mathbf{I}_{1+n \mod N}^{(N)}$ is obtained from the identity matrix as expressed in (27). The gradient estimation noise of the filtered-x sequential LMS algorithm at the minimum point, where the true gradient is zero, is given by

$$\mathbf{n}(n) = \hat{\nabla}(n) = -2e(n)x'(n).$$  (43)

Considering PU, only $L/N$ terms out of the $L$-length noise vector are nonzero at each iteration, giving a smaller noise contribution in comparison with the LMS algorithm, which updates the whole filter.

The weight vector covariance in the principal axis coordinate system, that is, in primed coordinates, is related to the covariance of the noise as follows:

$$\text{cov}[v'(n)] = \frac{G_p\mu}{8} \left( A - \frac{G_p\mu}{2} A^2 \right)^{-1} \text{cov}[n'(n)]$$

$$= \frac{G_p\mu}{8} \left( A - \frac{G_p\mu}{2} A^2 \right)^{-1} \text{cov}[Q^{-1}n(n)]$$

$$= \frac{G_p\mu}{8} \left( A - \frac{G_p\mu}{2} A^2 \right)^{-1} Q^{-1}E[n(n)n^T(n)]Q.$$  (44)

Assuming that $(G_p\mu/2)A$ is considerably less than $I$, then (44) simplifies to

$$\text{cov}[v'(n)] \approx \frac{G_p\mu}{8} A^{-1} Q^{-1} E[n(n)n^T(n)]Q.$$  (45)

The covariance of the gradient estimation error noise when the sequential PU is used can be expressed as

$$\text{cov}[n(n)] = E[n(n)n^T(n)]$$

$$= 4E[e^2(n)]\mathbf{I}_{1+n \mod N}^{(N)}x'(n)x'^T(n)\mathbf{I}_{1+n \mod N}^{(N)}$$

$$= 4E[e^2(n)]E[\mathbf{I}_{1+n \mod N}^{(N)}x'(n)x'^T(n)\mathbf{I}_{1+n \mod N}^{(N)}]$$

$$= 4E[e^2(n)]\frac{1}{N} \sum_{i=1}^{N} \mathbf{I}_{i}^{(N)}R\mathbf{I}_{i}^{(N)}$$

$$= 4E[e^2(n)]R_N.$$  (46)

In (46), statistical independence of the error and the input vector has been assumed at the minimum point of the error surface, where both signals are orthogonal.

According to (32), the comparison of (40) and (45)—carried out in terms of the trace of the autocorrelation matrices—confirms that the contribution of the gradient estimation noise is $N$ times weaker for the sequential LMS algorithm than for the LMS. This reduction compensates the eventual increase in the covariance of the weight vector in the principal axis coordinate system expressed in (45) when the maximum gain in step size $G_p = N$ is applied in the context of the $G_p$-FxSLMS algorithm.
4. EXPERIMENTAL RESULTS

In order to assess the effectiveness of the $G_{μ}$-FxSLMS algorithm, the proposed strategy was not only tested by simulation but was also evaluated in a practical DSP-based implementation. In both cases, the results confirmed the expected behavior: the performance of the system in terms of convergence rate and residual error is as good as the performance achieved by the FxLMS algorithm, even while the number of operations per iteration is significantly reduced due to PU.

4.1. Computer simulations

This section describes the results achieved by the $G_{μ}$-FxSLMS algorithm by means of a computer model developed in MATLAB on the theoretical basis of the previous sections. The model chosen for the computer simulation of the first example corresponds to the $1\times1\times1$ (1 reference microphone, 1 secondary source, and 1 error microphone) arrangement described in Figure 1(a). Transfer functions of the primary path $P(z)$ and secondary path $S(z)$ are shown in Figures 5(a) and 5(b), respectively. The filter modeling the primary path is a 64th-order FIR filter. The secondary path is modeled—by a 4th-order elliptic IIR filter—as a high pass filter whose cut-off frequency is imposed by the poor response of the loudspeakers at low frequencies. The offline estimate of the secondary path was carried out by an adaptive FIR filter of 200 coefficients updated by the LMS algorithm, as a classical problem of system identification. Figure 5(c) shows the transfer function of the estimated secondary path. The sampling frequency (8000 samples/s) as well as other parameters were chosen in order to obtain an approximate model of the real implementation. Finally, Figure 5(d) shows the power spectral density of $x(n)$, the reference signal for the undesired disturbance which has to be canceled

$$x(n) = \cos(2\pi 62.5n) + \cos(2\pi 187.5n) + \eta(n),$$

where $\eta(n)$ is an additive white Gaussian noise of zero mean whose power is

$$E[\eta^2(n)] = \sigma^2_{\eta} = 0.0001 \quad (-40 \text{ dB}).$$

After convergence has been achieved, the power of the residual error corresponds to the power of the random component of the undesired disturbance.

The length of the adaptive filter is of 256 coefficients. The simulation was carried out as follows: the step size was set to zero during the first 0.25 seconds; after that, it is set to 0.0001
and the adaptive process starts. The value $\mu = 0.0001$ is near the maximum stable step size when a decimation factor $N = 1$ is chosen.

The performance of the $G_\mu$-FxSLMS algorithm was tested for different values of the decimation factor $N$. Figure 6 shows the gain in step size over the frequency band of interest for different values of the parameter $N$. The gain in step size at the frequencies 62.5 Hz and 187.5 Hz are marked with two circles over the curves. The exact location of the notches is given by (22). On the basis of the position of the notches in the gain in step size and the spectral distribution of the undesired noise, the decimation factor $N = 64$ is expected to be critical because, according to Figure 6, the full-strength gain $G_\mu = N = 64$ cannot be applied at the frequencies 62.5 Hz and 187.5 Hz; both frequencies correspond exactly to the sinusoidal components of the periodic disturbance. Apart from the case $N = 64$ the gain in step size is free of notches at both of these frequencies.

Convergence curves for different values of the decimation factor $N$ are shown in Figure 7. The numbers that appear over the figures correspond to the mean-square error computed over the last 5000 iterations. The residual error is expressed in logarithmic scale as the ratio of the mean-square error and a signal of unitary power. As expected, the convergence rate and residual error are the same in all cases except when $N = 64$. For this value, the active noise control system diverges. In order to make the system converge when $N = 64$, it is necessary to decrease the gain in step size to a maximum value of 32 with a subsequent reduction in convergence rate.

The second example compares the theoretical gain in step size with the increase obtained by MATLAB simulation. The model of this example corresponds, as in the previous example, to the $1 \times 1 \times 1$ arrangement described in Figure 1. In this example, the reference is a single sinusoidal signal whose frequency varied in 20 Hz steps from 40 to 1560 Hz. The sampling frequency of the model is 3200 samples/s. Primary and secondary paths—$P(z)$ and $S(z)$—are pure delays of 300 and 40 samples, respectively. The output of the primary path is mixed with additive white Gaussian noise providing a signal-to-noise ratio of 27 dB. It is assumed that the secondary path has been exactly estimated. In order to provide very accurate results, the increase in step size between every two consecutive simulations looking for the bound is less than $1/5000$ the final value of the step size that ensures convergence. The

![Figure 6: Gain in step size over the frequency band of interest—from 0 to 400 Hz—for different values of the decimation factor $N$ ($N = 1, 2, 8, 32, 64, 80$).](image)
decimation factor $N$ of this example was set to 4. Figure 8 compares the predicted gain in step size with the achieved results. As expected, the experimental gain in step size is 4, apart from the notches that appear at 400, 800, and 1200 Hz.

### 4.2. Practical implementation

The $G_{\mu}$-FxSLMS algorithm was implemented in a $1 \times 2 \times 2$ active noise control system aimed at attenuating engine noise at the front seats of a Nissan Vanette. Figure 9 shows the physical arrangement of electroacoustic elements. The adaptive algorithm was developed on a hardware platform based on the DSP TMS320C6701 from Texas Instruments [18].

The length of the adaptive filter ($L$) for the $G_{\mu}$-FxSLMS algorithm was set to 256 or 512 coefficients (depending on the spectral characteristics of the undesired noise and the degree of attenuation desired), the length of the estimate of the secondary path ($L_s$) was set to 200 coefficients, and the decimation factor and the gain in step size were $N = G_{\mu} = 8$. The sampling frequency was $F_s = 8000$ samples/s. From the parameters selected, one can derive, according to (22), that the first notch in the gain in step size is located at 500 Hz.

The system effectively cancels the main harmonics of the engine noise. Considering that the loudspeakers have a low cut-off frequency of 60 Hz, the controller cannot attenuate the components below this frequency. Besides, the ANC system finds more difficulty in the attenuation of closely spaced frequency harmonics (see Figure 10(a)). This problem can be avoided by increasing the number of coefficients of the adaptive filter; for instance, from $L = 256$ to 512 coefficients (see Figure 10(b)).

In order to carry out a performance comparison of the $G_{\mu}$-FxSLMS algorithm with increasing value in the decimation term $N$—and subsequently in gain in step size $G_{\mu}$—it is essential to repeat the experiment with the same undesired disturbance. So to avoid inconsistencies in level and frequency, instead of starting the engine, we have previously recorded a signal consisting of several harmonics (100, 150, 200, and 250 Hz). An omnidirectional source (Brüel & Kjaer Omnipower 4296) placed inside the van is fed with this signal. Therefore, a comparison could be made under the same conditions. The ratio—in logarithmic scale—of the mean-square error and a signal of unitary power that appears over the graphics was calculated averaging the last
iterations shown. In this case, the length of the adaptive filter was set to 256 coefficients, the length of the estimate of the secondary path ($L_s$) was set to 200 coefficients, and the decimation factor and the gain in step size were set to $N = G_\mu = 1, 2, 4, \text{and } 8$. The sampling frequency was $F_s = 8000$ samples/s and the first notch in the gain in step size appeared at 500 Hz, well above the spectral location of the undesired disturbance. From the experimental results shown in Figure 11, the application of the full-strength gain in step size when the decimation factor is 2, 4, or 8 reduces the computational costs without degrading in any sense the performance of the system with respect to the full update algorithm.

Taking into account that the 2-channel ANC system implementing the $G_\mu$-FxSLMS algorithm inside the van ignored cross terms, the expressions given by Tables 1 and 2 show that approximately 32%, 48%, and 56% of the high-level multiplications can be saved when the decimation factor $N$ is set to 2, 4, and 8, respectively.

Although reductions in the number of operations are an indication of the computational efficiency of an algorithm, such reductions may not directly translate to a more efficient real-time DSP-based implementation on a hardware platform. To accurately gauge such issues, one must consider the freedoms and constraints that a platform imposes in the
harmonics at 100, 150, 200, and 250 Hz.

The system deals with a previously recorded signal consisting of harmonics at 100, 150, 200, and 250 Hz.

real implementation, such as parallel operations, addressing modes, registers available, or number of arithmetic units. In our case, the control strategy and the assembler code was developed trying to take full advantage of these aspects [5].

5. CONCLUSIONS

This work presents a contribution to the selection of the step size used in the sequential partial update LMS and FxLMS adaptive algorithms. The deterministic periodic input signal case is studied and it is verified that under certain conditions the stability range of the step size is increased compared to the full update LMS and FxLMS. The algorithm proposed here—filtered-x sequential LMS with gain in step size ($G_\mu$-FxSLMS)—is based on sequential PU of the coefficients of a filter and on a controlled increase in the step size of the adaptive algorithm. It can be used in active noise control systems focused on the attenuation of periodic disturbances to reduce the computational costs of the control system. It is theoretically and experimentally proved that the reduction of the computational complexity is not achieved at the expense of slowing down the convergence rate or of increasing the residual error.

The only condition that must be accomplished to take full advantage of the algorithm is that some frequencies should be avoided. These problematic frequencies correspond to notches that appear at the gain in step size. Their width and exact location depend on the system parameters.

Simulations and experimental results confirm the benefits of this strategy when it is applied in an active noise control system to attenuate periodic noise.

ACKNOWLEDGMENT

This work was partially supported by CICYT of Spanish Government under Grant TIN2005-08660-C04-01.

REFERENCES

[1] P. Lueg, “Process of silencing sound oscillations,” U.S. Patent no. 2.043.416, 1936.
[2] D. R. Morgan, “Analysis of multiple correlation cancellation loops with a filter in the auxiliary path,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 454–467, 1980.
[3] B. Widrow, D. Shur, and S. Shaffer, “On adaptive inverse control,” in *Proceedings of the 15th Asilomar Conference on Circuits, Systems, and Computers*, pp. 185–195, Pacific Grove, Calif, USA, November 1981.
[4] J. C. Burgess, “Active adaptive sound control in a duct: a computer simulation,” *Journal of the Acoustical Society of America*, vol. 70, no. 3, pp. 715–726, 1981.
[5] P. Ramos, R. Torrubia, A. López, A. Salinas, and E. Masgrau, “Computationally efficient implementation of an active noise control system based on partial updates,” in *Proceedings of the International Symposium on Active Control of Sound and Vibration (ACTIVE ‘04)*, Williamsburg, Va, USA, September 2004, paper 003.
[6] S. C. Douglas, “Adaptive filters employing partial updates,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 3, pp. 209–216, 1997.
[7] T. Aboulnasr and K. Mayyas, “Selective coefficient update of gradient-based adaptive algorithms,” in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ‘97)*, vol. 3, pp. 1929–1932, Munich, Germany, April 1997.
[8] K. Doğançay and O. Tanrikulu, “Adaptive filtering algorithms with selective partial updates,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 8, pp. 762–769, 2001.
[9] J. Sanubari, “Fast convergence LMS adaptive filters employing fuzzy partial updates,” in *Proceedings of IEEE Conference on Convergent Technologies for Asia-Pacific Region (TENCON ’03)*, vol. 4, pp. 1334–1337, Bangalore, India, October 2003.
[10] P. A. Naylor, J. Cui, and M. Brookes, “Adaptive algorithms for sparse echo cancellation,” *Signal Processing*, vol. 86, no. 6, pp. 1182–1192, 2006.
[11] S. M. Kuo, M. Tahernezhad, and W. Hao, “Convergence analysis of narrow-band active noise control system,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 2, pp. 220–223, 1999.
[12] E. Bjarnason, “Analysis of the filtered-X LMS algorithm,” *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 6, pp. 504–514, 1995.

[13] S. M. Kuo and D. R. Morgan, *Active Noise Control Systems: Algorithms and DSP Implementations*, John Wiley & Sons, New York, NY, USA, 1996.

[14] L. Vicente and E. Masgrau, “Novel FxLMS convergence condition with deterministic reference,” *IEEE Transactions on Signal Processing*, vol. 54, no. 10, pp. 3768–3774, 2006.

[15] S. J. Elliott, *Signal Processing for Active Control*, Academic Press, London, UK, 2001.

[16] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, USA, 1985.

[17] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Upper Saddle River, NJ, USA, 2002.

[18] Texas Instruments Digital Signal Processing Products, “TMS320C6000 CPU and Instruction Set Reference Guide,” 1999.
Research Article

Detection-Guided Fast Affine Projection Channel Estimator for Speech Applications

Yan Wu Jennifer, John Homer, Geert Rombouts, and Marc Moonen

1 Canberra Research Laboratory, National ICT Australia and Research School of Information Science and Engineering, The Australian National University, Canberra ACT 2612, Australia
2 School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane QLD 4072, Australia
3 Departement Elektrotechniek, Katholieke Universiteit Leuven, ESAT/SCD, Kasteelpark Arenberg 10, 3001 Heverlee, Belgium

Received 9 July 2006; Revised 16 November 2006; Accepted 18 February 2007

Recommended by Kutluyil Dogancay

In various adaptive estimation applications, such as acoustic echo cancellation within teleconferencing systems, the input signal is a highly correlated speech. This, in general, leads to extremely slow convergence of the NLMS adaptive FIR estimator. As a result, for such applications, the affine projection algorithm (APA) or the low-complexity version, the fast affine projection (FAP) algorithm, is commonly employed instead of the NLMS algorithm. In such applications, the signal propagation channel may have a relatively low-dimensional impulse response structure, that is, the number of active or significant taps within the (discrete-time modelled) channel impulse response is much less than the overall tap length of the channel impulse response. For such cases, we investigate the inclusion of an active-parameter detection-guided concept within the fast affine projection FIR channel estimator. Simulation results indicate that the proposed detection-guided fast affine projection channel estimator has improved convergence speed and has lead to better steady-state performance than the standard fast affine projection channel estimator, especially in the important case of highly correlated speech input signals.

Copyright © 2007 Yan Wu Jennifer et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

For many adaptive estimation applications, such as acoustic echo cancellation within teleconferencing systems, the input signal is highly correlated speech. For such applications, the standard normalized least-mean square (NLMS) adaptive FIR estimator suffers from extremely slow convergence. The use of the affine projection algorithm (APA) is considered as a modification to the standard NLMS estimators to greatly reduce this weakness. The built-in prewhitening properties of the APA greatly accelerate the convergence speed especially with highly correlated input signals. However, this comes with a significant increase in the computational cost. The lower complexity version of the APA, the fast affine projection (FAP) algorithm, which is functionally equivalent to APA, was introduced in [2].

The fast affine projection algorithm (FAP) is now, perhaps, the most commonly implemented adaptive algorithm for high correlation input signal applications.

For the above-mentioned applications, the signal propagation channels being estimated may have a “low dimensional” parametric representation [3–5]. For example, the impulse responses of many acoustic echo paths and communication channels have a “small” number of “active” (nonzero response) taps in comparison with the overall tap length of the adaptive FIR estimator. Conventionally, estimation of such low-dimensional channels is conducted using a standard FIR filter with the normalized least-mean square (NLMS) adaptive algorithm (or the unnormalized LMS equivalent). In these approaches, each and every FIR filter tap is NLMS-adapted during each time interval, which leads to relatively slow convergence rates and/or relatively poor steady-state performance. An alternative approach proposed by Homer et al. [6–8] is to detect and NLMS adapt only the active or significant filter taps. The hypothesis is that this can lead to improved convergence rates and/or steady-state performance.

Motivated by this, we propose the incorporation of an activity detection technique within the fast affine projection FIR channel estimator. Simulation results of the newly proposed detection-guided fast affine projection channel
2. SYSTEM DESCRIPTION

2.1. Adaptive estimator

We consider the adaptive FIR channel estimation system of Figure 1. The following assumptions are made:

1. All the signals are sampled: At sample instant \( k \), \( u(k) \) is the signal input to the unknown channel and the channel estimator; additive noise \( v(k) \) occurs within the unknown channel;
2. The unknown channel is linear and is adequately modeled by a discrete-time FIR filter \( \Theta = [\theta_0, \theta_1, \ldots, \theta_n]^T \) with a maximum delay of \( n \) sample intervals;
3. The additive noise signal is zero mean and uncorrelated with the input signal;
4. The FIR-modeled unknown channel, \( \Theta[z^{-1}] \) is sparsely active:

\[
\Theta[z^{-1}] = \theta_0 z^{-t_1} + \theta_1 z^{-t_2} + \cdots + \theta_m z^{-t_m},
\]

where \( m \ll n \), and \( 0 \leq t_1 < t_2 < \cdots < t_m \leq n \).

At sample instant \( k \), an active tap is defined as a tap corresponding to one of the \( m \) indices \( \{t_a\}_{a=1}^m \) of (1). Each of the remaining taps is defined as an inactive tap.

The observed output from the unknown channel is

\[
y(k) = \Theta^T U(k) + v(k),
\]

where \( U(k) = [u(k), u(k-1), \ldots, u(k-n)]^T \).

The standard adaptive NLMS estimator equation, as employed to provide an estimate \( \hat{\Theta} \) of the unknown channel impulse response vector \( \Theta \), is as follows [9]:

\[
\hat{\Theta}(k+1) = \hat{\Theta}(k) + \frac{\mu}{\delta + U^T(k)U(k)} [y(k) - \hat{y}(k)],
\]

where \( \hat{y}(k) = \hat{\Theta}(k)U(k) \) and where \( \delta \) is a small positive regularization constant.

Note: the standard initial channel estimate \( \hat{\Theta}(0) \) is the all-zero vector.

For stable 1st-order mean behavior, the step size \( \mu \) should satisfy \( 0 < \mu \leq 2 \). In practice, however, to attain higher-order stable behavior, the step size is chosen to satisfy \( 0 < \mu \ll 2 \).

For the standard discrete NLMS adaptive FIR estimator, every coefficient \( \hat{\theta}_i(k) \) \( i = 0, 1, \ldots, n \) is adapted at each sample interval. However, this approach leads to slow convergence rates when the required FIR filter tap length \( n \) is “large” [6]. In [6–8], it is shown that if only the active or significant channel taps are NLMS estimated then the convergence rate of the NLMS estimator may be greatly enhanced, particularly when \( m \ll n \).

2.2. Affine projection algorithm

The affine projection algorithm (APA) is considered as a generalisation of the normalized least-mean-square (NLMS) algorithm [2]. Alternatively, the APA can be viewed as an interchannel solution to the NLMS and RLS algorithms in terms of computational complexity and convergence rate [10]. The NLMS algorithm updates the estimator taps/weights on the basis of a single-input vector, which can be viewed as a one-dimensional affine projection [11]. In APA, the projections are made in multiple dimensions. The convergence rate of the estimator’s tap weight vector greatly increases with an increase in the projection dimension. This is due to the built-in decorrelation properties of the APA.

To describe the affine projection algorithm (APA) [1], the following notations are defined:
Hence, the ability problem of creating a singular matrix inverse when
\[ A(k) = \begin{bmatrix} U(k), U(k-1), \ldots, U(k-(N-1)) \end{bmatrix}, \]
where
\[ U(k) = \begin{bmatrix} u(k), u(k-1), \ldots, u(k-n) \end{bmatrix}^T; \]
\( \delta \): the adaptive estimator FIR tap weight vector at sample instant \( k \);
\( g(k) \): the channel estimation signal error vector of length \( N \);
\( y(k) \): system output;
\( v(k) \): the additive system noise;
\( \mu \): step size parameter.

The affine projection algorithm can be described by the following equations (see Figure 1).

The system output \( y(k) \) involves the channel impulse response to the excitation/input and the additive system noise \( v(k) \) and is given by (2).

The channel estimation signal error vector \( g(k) \) is calculated as
\[ g(k) = (Y(k) - U(k)\hat{\theta}(k-1), \]
where \( Y(k) = [y(k), y(k-1), \ldots, y(k-N+1)]^T. \)

The normalized residual channel estimation error vector \( \hat{g}(k) \) is calculated in the following way:
\[ \hat{g}(k) = (U(k) - U(k)\hat{\theta}(k) + \delta I)^{-1}. \]
where \( I = N \times N \) identity matrix.

The APA channel estimation vector is updated in the following way:
\[ \hat{\theta}(k+1) = \hat{\theta}(k) + U(k)\hat{g}(k). \]
A regularization term \( \delta \) times the identity matrix is added to the covariance matrix within (5) to prevent the instability problem of creating a singular matrix inverse when \( [U(k)^T - U(k)] \) has eigenvalues close to zero. A well behaved inverse will be provided if \( \delta \) is large enough.

From the above equations, it is obvious that the relations (4), (5), (6) reduce to the standard NLMS algorithm if \( N = 1 \). Hence, the affine projection algorithm (APA) is a generalization of the NLMS algorithm.

### 2.3 Fast affine projection algorithm

The complexity of the APA is about \( 2(n+1)N + 7N^2 \), which is generally much larger than the complexity of the NLMS algorithm, \( 2(n+1) \). Motivated by this, a fast version of the APA was derived in [2]. Here, instead of calculating the error vector from the whole covariance matrix, the FAP only calculates the first element of the \( N \)-element error vector, where an approximation is made for the second to the last components of the error vector \( e(k) \) as \( (1 - \mu) \) times the previously computed error [12, 13]:

\[ g(k+1) = \left[ \begin{array}{c} e(k+1) \\ (1-\mu)\tilde{e}(k) \end{array} \right], \]

where the \( N - 1 \) length vector consists of the \( N - 1 \) upper elements of the vector \( g(k) \).

Note: (7) is an exact formula for the APA if and only if \( \delta = 0 \).

The second complexity reduction is achieved by only adding a weighted version of the last column of \( U(k) \) to update the tap weight vector. Hence there are just \( (n+1) \) multiplications as opposed to \( N \times (n+1) \) multiplications for the APA update of (6). Here, an alternate tap weight vector \( \hat{\theta}_1(k) \) is introduced.

Note: the subscript 1 denotes the new calculation method.

\[ \hat{\theta}_1(k+1) = \hat{\theta}_1(k) - \mu U(k-N+2)E_{N-1}(k+1), \]

where

\[ E_{N-1}(k+1) = \sum_{j=0}^{N-1} e_j(k-N+2+j) \]

is the \( (N-1) \)th element in the vector

\[ e(k+1) = \left[ \begin{array}{c} e_0(k+1) \\ e_1(k+1) + e_0(k) \\ \vdots \\ e_{N-1}(k+1) + e_{N-2}(k) + \cdots + e_0(k-N+2) \end{array} \right]. \]

Alternatively, \( e(k+1) \) can be written as

\[ E(k+1) = \left[ \begin{array}{c} 0 \\ F(k) \end{array} \right] + g(k+1), \]

where \( F(k) \) is an \( N - 1 \) length vector consisting of the upper most \( N - 1 \) elements of \( E(k) \) and \( g(k+1) \) as calculated via (5).

Hence, it can be shown that the relationship between the new update method and the old update method of APA can be viewed as

\[ \hat{\theta}(k) = \hat{\theta}_1(k) + \mu U(k)\hat{F}(k), \]

where \( U(k) \) consists of the \( N - 1 \) leftmost columns of \( U(k) \).
A new efficient method to calculate $e(k)$ using $\hat{\theta}_1(k)$ rather than $\hat{\theta}(k)$ is also derived:

$$\tilde{r}_{xx}(k+1) = \tilde{r}_{xx}(k) + u(k+1)\tilde{\alpha}(k+1) - u(k-n)\tilde{\alpha}(k-n),$$

(13)

where

$$\tilde{\alpha}(k+1) = [u(k), u(k-1), \ldots, u(k-N+2)]^T$$

(14)

$$e_1(k+1) = y(k+1) - \tilde{U}(k+1)^T\hat{\theta}_1(k)$$

(15)

$$e(k+1) = e_1(k+1) - \mu \tilde{r}_{xx}(k+1)\tilde{\beta}(k).$$

(16)

(Further details can be found in [2].)

The following is a summary of the FAP algorithm:

1. $\tilde{r}_{xx}(k+1) = \tilde{r}_{xx}(k) + u(k+1)\tilde{\alpha}(k+1) - u(k-n)\tilde{\alpha}(k-n)$,
2. $e_1(k+1) = y(k+1) - \tilde{U}(k+1)^T\hat{\theta}_1(k)$,
3. $e(k+1) = e_1(k+1) - \mu \tilde{r}_{xx}(k+1)\tilde{\beta}(k)$,
4. $g(k+1) = \left[ e(k+1) \right]_{(1-\mu[\tilde{\alpha}]k)}$,
5. $g(k+1) = [\tilde{U}(k+1)^T\tilde{U}(k+1) + \delta I]^{-1}g(k+1)$,
6. $\tilde{\beta}(k+1) = \left[ \frac{0}{\tilde{\beta}(k)} \right] + g(k+1)$,
7. $\hat{\theta}_1(k+1) = \hat{\theta}_1(k) - \mu \tilde{U}(k-N+2)E_{-1}(k+1)$.

The above formulae are in general only approximately equivalent to the APA; they are exactly equal to the APA if the regularization $\delta$ is zero. Steps (2) and (7) of the FAP algorithm are each of complexity $(N+1)$ MPSI (multiplications per symbol interval). Step (1) is of complexity $2N$ MPSI and steps (3), (4), (6) are each of complexity $N$ MPSI. Step (5), when implemented in the Levinson-Dubin method, requires $7N^2$ MPSI [2]. Thus, the complexity of FAP is roughly $2(N+1) + 7N^2 + 5N$. For many applications like echo cancellation, the filter length $(N+1)$ is always much larger than the required affine projection order $N$, which makes FAP’s complexity comparable to that of NLMS. Furthermore, the FAP only requires slightly more memory than the NLMS.

3. DETECTION-GUIDED ESTIMATION

3.1. Least-squares activity detection criteria review

The original least-squares-based detection criterion for identifying active FIR channel taps for white input signal conditions [6] is as follows.

The tap index $j$ is defined to be detected as a member of the active tap set $\{t_{a}\}_{m=1}^{n}$ at sample instant $k$ if

$$X_j(k) > T(k),$$

(17)

where

$$X_j(k) = \frac{\sum_{i=1}^{k} [y(i)u(i-j)]^2}{\sum_{i=1}^{k} u^2(i-j)},$$

$$T(k) = \frac{2 \log(k)}{k} \sum_{i=1}^{k} y^2(i).$$

(18)

However, the original least-square-based detection criterion suffers from tap coupling problems when colored or correlated input signals are applied. In particular, the input correlation causes $X_j(k)$ to depend not only on $\theta_j$ but also the neighboring taps.

The following three modifications to the above activity detection criterion were proposed in [7, 8] for providing enhanced performance for applications involving nonwhite input signals.

Modification 1. Replace $X_j(k)$ by

$$\tilde{X}_j(k) = \frac{\sum_{i=1}^{k} [y(i) - \hat{y}(i) + \theta_j(i)u(i-j)]u(i-j)]^2}{\sum_{i=1}^{k} u^2(i-j)}.$$  

(19)

The additional term $-\hat{y}(i) + \theta_j(i)u(i-j)$ in the numerator of $\tilde{X}_j(k)$ is used to reduce the coupling between the neighboring taps [7, 8].

Modification 2. Replace $T(k)$ by

$$\tilde{T}(k) = \frac{2 \log(k)}{k} \sum_{i=1}^{k} [y(i) - \hat{y}(i)]^2.$$  

(20)

This modification is based on the realization that for inactive taps, the numerator term of $\tilde{X}_j(k)$ is approximately

$$N_j(k) \approx \left\{ \sum_{i=1}^{k} [y(i) - \hat{y}(i)]u(i-j) \right\}^2, \quad j = \text{inactive tap index}.$$  

(21)

Combining this with the LS theory on which the original activity criterion (17) is based suggests the following modification [8].

Modification 3. Apply an exponential forgetting operator $W_k(i) = (1-y)^{k-1}, 0 < y \ll 1$ within the summation terms of the activity criterion [8].

Modification 2 is theoretically correct only if $\Theta - \hat{\Theta}(k)$ is not time varying. Clearly this is not the case. Modification 3 is included to reduce the effect of $\Theta - \hat{\Theta}(k)$ being time varying. Importantly, the inclusion of Modification 3 also improves the applicability of the detection-guided estimator to time-varying systems. (Note that the result of Modification 3 is not time varying.)

3.2. Enhanced detection-guided NLMS FIR channel estimator

The enhanced time-varying detection-guided NLMS estimation proposed in [8] is as follows.

For each tap index $j$ and at each sample interval:

1. label the tap index $j$ to be a member of the active parameter set $\{t_{a}\}_{m=1}^{n}$ at sample instant $k$ if

$$\tilde{X}_j^w(k) > \tilde{T}^w(k),$$

(22)
where
\[ \hat{X}_j^w(k) = \frac{\sum_{i=1}^k W_k(i) [y(i) - \hat{y}(i)] u(i - j)]^2}{\sum_{i=1}^k W_k(i) u^2(i - j)}, \]
(23)
\[ \tilde{W}_w(k) = \frac{2 \log (L_w(k))}{L_w(k)} \sum_{i=1}^k W_k(i) [(y(i) - \hat{y}(i))]^2, \]
(24)
\[ L_w(k) = \sum_{i=1}^k W_k(i), \]
(25)
and where \( W_k(i) \) is the exponentially decay operator:
\[ W_k(i) = (1 - y)^{k-i}, \]
(26)
(2) update the NLMS weight for each detected active tap index \( t_a \):
\[ \hat{\theta}_a(k + 1) = \hat{\theta}_a(k) + \frac{\mu}{\sum_{t_a} u(k - t_a)^2 + \epsilon} u(k - t_a) e(k), \]
(27)
where \( \sum_{t_a} \) summation over all detected active-parameter indices;
(3) reset the NLMS weight to zero for each identified inactive tap index.

Note that (23)–(25) can be implemented in the following recursive form:
\[ N_j(k) = (1 - y) N_j(k-1) + [y(k) - \hat{y}(k) + \hat{\theta}_j(k) u(k-j)] u(k-j), \]
\[ D_j(k) = (1 - y) D_j(k-1) + u^2(k-j), \]
\[ q(k) = (1 - y) q(k-1) + [y(k) - \hat{y}(k)]^2, \]
\[ L^w(k) = (1 - y) L^w(k-1) + 1, \]
\[ \tilde{X}_j^w(k) = \frac{N_j^w(k)}{D_j^w(k)}, \]
\[ \tilde{T}^w(k) = \frac{2q(k) \log [L^w(k)]}{L^w(k)}. \]
(29)

Note, as suggested in [8], that a threshold scaling constant \( \eta \) may be introduced on the right-hand side of (24) or (29). If \( \eta > 1 \), the system may avoid the incorrect detection of “non-active” taps. This, however, may come with an initial delay in detecting the smallest of the active taps, leading to an initial additional error increase. If \( \eta < 1 \), it may improve the detectability of “weak” active taps. However, it has the risk of incorrectly including inactive taps within the active tap set, resulting in reduced convergence rates.

### 3.3. **Proposed detection-guided FAP FIR channel estimator**

The enhanced detection-guided FAP estimation is derived as follows.

The tap index \( j \) is detected as being a member of the active parameter set \( \{ t_a \}_{a=1}^m \) at sample instant \( k \) if
\[ \tilde{X}_j^w(k) > \tilde{T}^w(k), \]
(30)
where
\[ \tilde{X}_j^w(k) = \frac{\sum_{i=1}^k W_k(i) [e_j(i) + \hat{\theta}_j(i) u(i-j)] u(i-j)]^2}{\sum_{i=1}^k W_k(i) u^2(i-j)}, \]
(31)
\[ \tilde{T}^w(k) = \frac{2 \log (L_w(k))}{L_w(k)} \sum_{i=1}^k W_k(i) [(e_j(i))]^2, \]
(32)
\[ L_w(k) = \sum_{i=1}^k W_k(i), \]
(33)
and where \( W_k(i) \) is the exponentially decay operator
\[ W_k(i) = (1 - y)^{k-i}, \]
(34)
and \( \hat{\theta}_j(i) \) is the \( j \)th element of \( \hat{\theta}_j \) as defined in (8), (11), and \( e_j(i) \) as defined in (15).

We propose to apply this active detection criterion to the fast affine projection algorithm. This involves creating an \((n + 1) \times (n + 1)\) diagonal activity matrix \( B(k) \), where the \( j \)th diagonal element \( B_j(k) = 1 \) if the \( j \)th tap index is detected as being active at sample instant \( k \), otherwise \( B_j(k) = 0 \). This matrix is then applied within the FAP algorithm as follows.

Replace (5) with
\[ \varepsilon_d(k) = [(B(k) U(k))^T [B(k) U(k)] + \delta I]^{-1} \varepsilon(k). \]
(35)

Replace (11) with
\[ \tilde{E}_d(k) = \begin{bmatrix} 0 \\ \tilde{E}_d(k-1) \end{bmatrix} + \varepsilon_d(k). \]
(36)

Replace (8) with
\[ \hat{\theta}_d(k) = B(k) \hat{\theta}_d(k-1) - \mu B(k) U(k-N+1) \tilde{E}_{d,N-1}(k), \]
(37)
where
\[ \tilde{E}_{d,N-1}(k) = \sum_{j=0}^{N-1} \varepsilon_{d,j} (k-N+1+j) \]
(38)
and \( \varepsilon_{d,j} \) is the \( j \)th element of \( \varepsilon_d \).

As with the detection-guided NLMS algorithm, a threshold scaling constant \( \eta \) may be introduced on the right-hand side of (32) based on different conditions. The effectiveness of this scaling constant is considered in the simulations.

### 3.4. **Computational complexity**

The proposed system requires \( 4(n + 1) + 4 \) MPSI to perform the detection tasks required in the recursive equivalent of (30)–(33). By including the sparse diagonal matrix \( B(k) \) in (37), the system only needs to include \( m \) multiplications rather than \((n + 1)\) multiplications for (15) and (8). Thus, the proposed detection-guided FAP channel estimator requires \( 2m + 7N^2 + 5N + 4(n+1) \) + 4 MPSI while the complexity of FAP is \( 2(n + 1) + 7N^2 + 5N \) MPSI. Hence, for sufficiently long, low-dimensional active channels \( n \gg m \geq 1, n \gg N \), the computational cost of the proposed detection-guided FAP channel estimator is essentially twice that of the FAP and of the standard NLMS estimators.
4. SIMULATIONS

Simulations were carried out to investigate the performance of the following channel estimators when different input signals with different correlation levels are applied.

(A) Standard NLMS channel estimator.
(B) Active-parameter detection-guided NLMS channel estimator (as presented in Section 3.2).
(C) APA channel estimator with \( N = 10 \).
(D) FAP channel estimator with \( N = 10 \).
(E) Active-parameter detection-guided FAP channel estimator with \( N = 10 \) (without threshold scaling).
(F) Active-parameter detection-guided FAP channel estimator with \( N = 10 \), with threshold scaling constant.
(G) FAP channel estimator with \( N = 14 \). In this case, it has almost the same computational complexity\(^1\) as that of the active-parameter detection-guided FAP channel estimator with \( N = 10 \).

Simulation conditions are the following.

(a) The channel impulse response considered, as given in Figure 2(a), was based on a real acoustic echo channel measurement made by CSIRO Radiophysics, Sydney, Australia. The impulse response of Figure 2(a) was derived from a measured acoustic echo path impulse response, Figure 2(b), by applying the technique based on the Dohono thresholding principle\(^2\), as presented in the appendix. This technique essentially removes the effects of estimation/measurement noise. The measured impulse response of Figure 2(b) was obtained from a room approximately \( 5 \times 10 \times 3 \) m.

The noise thresholded impulse response of Figure 2(a) consists of \( m = 11 \) active taps and a total tap length of \( n = 300 \).

The channel response used in the simulations is an example of a room acoustic impulse response which displays a sparse-like structure. Note, whether or not a room acoustic impulse response is sparse-like depends on the room configuration (size, placement of furniture, wall/floor coverings, microphone and speaker positioning). Nevertheless, a significant proportion of room acoustic impulse responses are, to varying degrees, sparse-like.

(b) Adaptive step size \( \mu = 0.005 \).
(c) Regularization parameter \( \delta = 0.1 \).
(d) Initial channel estimate \( \hat{\theta}(0) \) is the all-zero vector.
(e) Noise signal \( \nu(k) = \) zero mean Gaussian process with variance of either \( 0.01 \) (Simulations 1 to 3) or \( 0.05 \) (Simulation 4).
(f) The squared channel estimator error \( ||\theta - \hat{\theta}||^2 \) is plotted to compare the convergence rate. All plots are the average of 10 similar simulations.
(g) For the simulations of the detection-guided NLMS channel estimator and the detection-guided FAP channel estimator, the forgetting parameter \( \gamma = 0.001 \).

Simulation 1. Lowly correlated coloured input signal \( u(k) \) described by the model \( u(k) = w(k)/[1 - 0.1z^{-1}] \), where \( w(k) \) is a discrete white Gaussian process with zero mean and unit variance.

Simulation 2. Highly correlated input signal \( u(k) \) described by the model \( u(k) = w(k)/[1 - 0.9z^{-1}] \), where \( w(k) \) is a discrete white Gaussian process with zero mean and unit variance.

\(^1\) The complexity is calculated based on the discussion in Section 3.4. The computational complexity of the active-parameter detection-guided FAP channel estimator with \( N = 10 \) is 1980 MPSI, which is slightly lower than the complexity of standard FAP with \( N = 14 \) of 2044 MPSI.

\(^2\) Theorem 3.4 of Ref. 14.
**Simulation 3.** Tenth-order AR-modelled speech input signal.

**Simulation 4.** Tenth-order AR-modelled speech input signal under noisy conditions. That is, with higher noise variance = 0.05.

In all four simulations, two detection-guided scaling constants were employed: $\eta = 1$ (i.e., no scaling) and $\eta = 4$.

5. **RESULT AND ANALYSIS**

**Simulation 1** (lowly correlated input signal case). The results of the simulations for channel estimators (a) to (g) with $\mu = 0.005$ are shown in Figure 3.

(a) Channel estimators (b) to (f) show faster convergence than the standard NLMS channel estimator (a).

(b) The detection-guided NLMS estimator (b) provides faster convergence rate than the APA channel estimator (c) with $N = 10$ and the FAP channel estimator (d) with $N = 10$. It is clear that the APA channel estimator (c) with $N = 10$ and FAP channel estimator (d) with $N = 10$ still have not reached steady state at the 20000 sample mark.

(c) The detection-guided FAP channel estimators with $N = 10$ (e), (f) show a better convergence rate than channel estimators (b), (c), and (d).

(d) Detection-guided FAP estimator (e) with detection-guided FAP estimator with threshold scaling constant $\eta = 4$ (f) both can detect all the active taps and almost have the same performance.

(e) With almost the same computational cost, detection-guided FAP estimator (e) significantly outperforms standard FAP estimator with $N = 14$ in terms of convergence rate.

**Simulation 2** (highly correlated input signal case). The results of the simulations for channel estimators (a) to (g) with $\mu = 0.005$ are shown in Figure 4.

(a) The active-parameter detection-guided NLMS channel estimator (b) does not provide suitably enhanced improved convergence speed over the standard NLMS channel estimator (a). This is due to the incorrect detection of many of the inactive taps with the highly correlated input signals.

(b) The APA channel estimator with $N = 10$ (c) and the FAP channel estimator with $N = 10$ (d) show significantly improved convergence over (a) and (b). This is due to the autocorrelation matrix inverse $[U(k)^T U(k) + \delta I]^{-1}$ in (5) essentially prewhitening the highly colored input signal.

(c) The detection-guided FAP channel estimators with $N = 10$ (e), (f) show better convergence rates than the standard APA channel estimator with $N = 10$ (c) and the standard FAP channel estimator with $N = 10$ (d). In addition, the detection-guided FAP estimators (e), (f) appear to provide better steady-state error performance.

(d) The detection-guided FAP channel estimator (e) without threshold scaling detects extra “nonactive” taps. In the simulation, it detects 32 active taps, which are 21 in excess of the true number. This leads to slower convergence rate. In comparison, the detection-guided FAP channel estimator (f) with threshold scaling $\eta = 4$, it shows the ability to detect the correct number of active taps, however, this comes with a relative initial error increase.

(e) The detection-guided FAP channel estimator (e) with $N = 10$ provides noticeably better convergence rate performance than the standard FAP channel estimator (d) with $N = 14$ in terms of the convergence rate and the steady-state error.

**Simulation 3** (highly correlated speech input signal case). The results of the simulations for channel estimators (a) to (g) with $\mu = 0.005$ are shown in Figure 5. The trends shown here are similar to those of Simulations 1 and 2, although here the convergence rate and steady-state benefits provided by detection guiding are further accentuated.

(a) When the speech input signal is applied, the active parameter detection-guided NLMS channel estimator (b) suffers from very slow convergence, similar to that of the standard NLMS channel estimator (a). This is due to the incorrect detection of many of the inactive taps.

(b) The detection-guided FAP channel estimators (e) and (f) significantly outperform channel estimators (c) and (d) in terms of convergence speed. The results also indicate that the newly proposed detection-guided FAP estimators may have better steady state error performance than the standard APA and FAP estimators.

(c) For detection FAP estimator (e) and detection FAP estimator with threshold scaling constant $\eta = 4$ (f), the trends are similar to those observed for Simulation 2: detection FAP estimator (e) detects extra 23 active taps, resulting in reduced convergence rate and there is an initial error increase occurring in detection FAP estimator with threshold scaling constant $\eta = 4$ (f).

(d) Again, with the same computational cost, the detection-guided FAP channel estimator (e) with $N = 10$ shows a faster convergence rate and reduced steady state error relative to standard FAP channel estimator (d) with $N = 14$.

**Simulation 4** (highly correlated speech input signal case with higher noise variance). The results of the simulations for channel estimators (a) to (g) with $\mu = 0.005$ are shown in Figure 6, which confirm the similar good performance of our newly proposed channel estimator under noisy conditions. The detection FAP estimator with threshold scaling constant $\eta = 4$ (f) performs noticeably better than the detection estimator FAP without threshold scaling (e) due to the ability to detect the correct number of active taps.
Figure 3: Comparison of convergence rates for lowly correlated input signal.
Figure 4: Comparison of convergence rates for highly correlated input signal.
Figure 5: Comparison of convergence rates for speech input signal.
Figure 6: Comparison of convergence rates for speech input signal under noisy conditions.
6. CONCLUSION

For many adaptive estimation applications, such as acoustic echo cancellation within teleconferencing systems, the input signal is speech or highly correlated. In such applications, the standard NLMS channel estimator suffers from extremely slow convergence. To remove this weakness, the affine projection algorithm (APA) or the related computationally efficient fast affine projection (FAP) algorithm is commonly employed instead of the NLMS algorithm. Due to the signal propagation channels in such applications, sometimes having low dimensional or sparsely active impulse responses, we considered the incorporation of active-parameter detection with the FAP channel estimator. This newly proposed detection-guided FAP channel estimator is characterized with improved convergence speed and perhaps also better steady-state error performance as compared to the standard FAP estimator. The similar good performance is also achieved under noisy conditions. Additionally, simulations confirm these advantages of the proposed channel estimator under essentially the same computational cost. These features make this newly proposed channel estimator a good candidate for the adaptive estimation speech applications such as the acoustic echo cancellation problem.

APPENDICES

A. SPARSE CHANNEL IMPULSE RESPONSE ESTIMATION: REMOVING MEASUREMENT NOISE EFFECTS

In this appendix, a procedure for removing the measurements noise effect from the estimated time domain channel impulse response is presented. This procedure may be viewed as an offline scheme for active-tap detection of sparse channels and assumes that the true impulse response has a sufficiently large number of zero taps. Its applicability is restricted to channels which have a sparse structure.

In general, the presence of measurement noise or disturbance causes the tap coefficients of each of the zero taps of the sparse channel to be nonzero. If we assume the estimate was obtained with a white input, then the discussion of Section 3 (more details can be found in [15]) suggests that asymptotically (at least for LS, LMS estimates) the zero-tap estimates have a zero mean i.i.d Gaussian distribution:

$$\hat{\theta}_i \sim N(0, \sigma^2), \text{ i.i.d, where } \hat{\theta}_i = 0.$$  \hspace{1cm} (A.1)

Under the validity of (A.1), we use the following results from the work of Donoho cited in [15], to develop a procedure for removing the effects of the noise, or, equivalently, for determining which taps are zero.

B. RESULT

Let $$\{\hat{\theta}_i\} \sim N(0, \sigma^2), \text{ i.i.d. Define the event } A_M = \{\sup_{i=1:M} |z_i| \leq \sigma \sqrt{2 \log M}\}, \text{ Then, } \text{Prob}(A_M) \rightarrow 1 \text{ as } M \rightarrow \infty.$$  \hspace{1cm} (B.1)

A priori knowledge of the indices i of the zero taps is required in order to use the threshold $$\sigma \sqrt{2 \log M}$$ to determine which taps are zero. By applying the following iterative procedure, this requirement is avoided for sparse channels.

Algorithm 1. (1) Initially, include the indices of all n tap estimates $$\{\hat{\theta}_i\}$$ in the set $$S$$ of zero taps and set $$M = n$$.

(2) Determine rms value $$\sigma_S$$ of the estimates in Set $$S$$.

(3) Determine the indices i of those taps for which the estimates coefficients satisfy

$$|\hat{\theta}_i| \leq \sigma_S \sqrt{2 \log M}.$$  \hspace{1cm} (B.1)

(4) Repeat steps (2) and (3) a given number of times or, alternatively, until the difference in $$\sigma_S$$ from one iteration to the next has decreased to a given value.

ACKNOWLEDGMENT

The authors would like to acknowledge CSIRO Rdiophysics, Sydney for providing the measurement data of the simulation channel.

REFERENCES

[1] K. Ozeki and T. Umeda, “An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties,” Electronics & Communications in Japan, vol. 67, no. 5, pp. 19–27, 1984.
[2] S. L. Gay and S. Tavathia, “The fast affine projection algorithm,” in Proceedings of the 20th International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’95), vol. 5, pp. 3023–3026, Detroit, Mich, USA, May 1995.
[3] J. R. Casar-Corredera and J. Alcazar-Fernandez, “An acoustic echo canceller for teleconference systems,” in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’86), vol. 11, pp. 1317–1320, Tokyo, Japan, April 1986.
[4] A. Gilloire and J. Zurcher, “Achieving the control of the acoustic echo in audio terminals,” in Proceedings of European Signal Processing Conference (EUSIPCO ‘88), pp. 491–494, Grenoble, France, September 1988.
[5] S. Makino and S. Shimada, “Echo control in telecommunication,” Journal of the Acoustic Society of Japan, vol. 11, no. 6, pp. 309–316, 1990.
[6] J. Homer, I. Mareels, R. R. Bitmead, B. Wahlberg, and A. Gustafsson, “LMS estimation via structural detection,” IEEE Transactions on Signal Processing, vol. 46, no. 10, pp. 2651–2663, 1998.
[7] J. Homer, “Detection guided NLMS estimation of sparsely parametrized channels,” IEEE Transactions on Circuits and Systems II, vol. 47, no. 12, pp. 1437–1442, 2000.
[8] J. Homer, I. Mareels, and C. Hoang, “Enhanced detection-guided NLMS estimation of sparse FIR-modeled signal channels,” IEEE Transactions on Circuits and Systems I, vol. 53, no. 8, pp. 1783–1791, 2006.
[9] S. Haykin, Adaptive Filter Theory, Prentice Hall Information and System Science Series, Prentice-Hall, Upper Saddle River, NJ, USA, 3rd edition, 1996.
[10] M. Bouchard, “Multichannel affine and fast affine projection algorithms for active noise control and acoustic equalization systems,” IEEE Transactions on Speech and Audio Processing, vol. 11, no. 1, pp. 54–60, 2003.
[11] S. G. Sankaran and A. A. Beex, “Convergence behavior of affine projection algorithms,” *IEEE Transactions on Signal Processing*, vol. 48, no. 4, pp. 1086–1096, 2000.

[12] G. Rombouts and M. Moonen, “A sparse block exact affine projection algorithm,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 2, pp. 100–108, 2002.

[13] G. Rombouts and M. Moonen, “A fast exact frequency domain implementation of the exponentially windowed affine projection algorithm,” in *Proceedings of IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium (AS-SPCC ’00)*, pp. 342–346, Lake Louise, Alta., Canada, October 2000.

[14] M. R. Leadbetter, G. Lindgren, and H. Rootzen, *Extremes and Related Properties of Random Sequences and Processes*, Springer, New York, NY, USA, 1982.

[15] H. Cramer and M. R. Leadbetter, *Stationary and Related Stochastic Processes: Sample Function Properties and Their Applications*, John Wiley & Sons, New York, NY, USA, 1967.
Research Article

Efficient Multichannel NLMS Implementation for Acoustic Echo Cancellation

Fredric Lindstrom,¹ Christian Schüldt,² and Ingvar Claesson²

¹ Konftel AB, Research and Development, Box 268, 90106 Umeå, Sweden
² Department of Signal Processing, Blekinge Institute of Technology, 37225 Ronneby, Sweden

Received 31 May 2006; Revised 9 November 2006; Accepted 14 November 2006

Recommended by Kutluyil Dogancay

An acoustic echo cancellation structure with a single loudspeaker and multiple microphones is, from a system identification perspective, generally modelled as a single-input multiple-output system. Such a system thus implies specific echo-path models (adaptive filters) for every loudspeaker to microphone path. Due to the often large dimensionality of the filters, which is required to model rooms with standard reverberation time, the adaptation process can be computationally demanding. This paper presents a selective updating normalized least mean square (NLMS)-based method which reduces complexity to nearly half in practical situations, while showing superior convergence speed performance as compared to conventional complexity reduction schemes. Moreover, the method concentrates the filter adaptation to the filter which is most misadjusted, which is a typically desired feature.

Copyright © 2007 Fredric Lindstrom et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Acoustic echo cancellation (AEC) [1, 2] is used in teleconferencing equipment in order to provide high quality full-duplex communication. The core of an AEC solution is an adaptive filter which estimates the impulse response of the loudspeaker enclosure microphone (LEM) system. Typical adaptive algorithms for the filter update procedure in the AEC are the least mean square, normalized least mean square (LMS, NLMS) [3], affine projection (AP), and recursive least squares (RLS) algorithms [4]. Of these, the NLMS-based algorithms are popular in industrial implementations, thanks to their low complexity and finite precision robustness.

Multimicrophone solutions are frequent in teleconferencing equipment targeted for larger conference rooms. This paper considers a system consisting of one loudspeaker and three microphones. The base unit of the system contains the loudspeaker and one microphone and it is connected to two auxiliary expansion microphones, as shown in Figure 1. Such multimicrophone system constitutes a single-input multiple-output (SIMO) multichannel system with several system impulse responses to be identified, Figure 2. Thus, the signal processing task can be quite computational demanding.

Several methods for computational complexity reduction of the LMS/NLMS algorithms have been proposed and analyzed, for example, [5–14]. In this paper a related low complexity algorithm for use in a multimicrophone system is proposed.

2. COMPLEXITY REDUCTION METHODS

The LEM system can be modelled as a time invariant linear system, \( \mathbf{h}(k) = [h_0(k), \ldots, h_{N-1}(k)]^T \), where \( N - 1 \) is the order of the finite impulse response (FIR) model [11] and \( k \) is the sample index. Thus, the desired (acoustic echo) signal \( d(k) \) is given by \( d(k) = \mathbf{h}(k)^T \mathbf{x}(k) \), where \( \mathbf{x}(k) = [x(k), \ldots, x(k-N+1)]^T \) and \( x(k) \) is the input (loudspeaker) signal. The measured (microphone) signal \( y(k) \) is obtained as \( y(k) = d(k) + n(k) \), where \( n(k) \) is near-end noise. Assuming an adaptive filter \( \mathbf{h}(k) \) of length \( N \) is used, that is, \( \hat{\mathbf{h}}(k) = [\hat{h}_0(k), \ldots, \hat{h}_{N-1}(k)]^T \), the NLMS algorithm is given by

\[
\begin{align*}
e(k) & = y(k) - \hat{d}(k) = y(k) - \mathbf{x}(k)^T \hat{\mathbf{h}}(k), \\
\beta(k) & = \frac{\mu}{||\mathbf{x}(k)||^2 + \epsilon}, \\
\hat{\mathbf{h}}(k+1) & = \hat{\mathbf{h}}(k) + \beta(k)e(k)\mathbf{x}(k),
\end{align*}
\]

(1)

(2)
where $\hat{d}(k)$ is the estimated echo, $e(k)$ the error (echo cancelled) signal, $\beta(k)$ the step-size, $\|x(k)\|^2 = x(k)^T x(k)$ the squared Euclidian norm, $\mu$ the step size control parameter, and $\epsilon$ a regularization parameter [4].

Low-complexity periodical and partial updating schemes reduce the computational complexity of the LMS/NLMS by performing only a part of the filtering update, (2). The periodic NLMS performs the filter update only at periodical sample intervals. This updating can be distributed over the intermediate samples [5]. The sequential NLMS updates only a part of the $N$ coefficients at every sample in a sequential manner [5]. Several methods for choosing which coefficients to update at what sample instant have been proposed, for example, choosing a subset containing the largest coefficients in the regressor vector [6], low-complexity version of largest regressor vector coefficient selection [7], block-based regressor vector methods [8, 9], and schemes based on randomization in the update procedure [10]. The updating can also be based on assumptions of the unknown plant [11, 12]. Another approach of omitting updates is possible in algorithms where the step size is zero for a large number of updates [13, 14].

In a SIMO-modelled $M$ microphone system, there are $M$ adaptive filters $h_m(k)$ with $m \in \{1, \ldots, M\}$, to be updated at each sample, that is,

$$
\hat{h}_m(k+1) = \hat{h}_m(k) + \frac{\mu e_m(k) x(k)}{\|x(k)\|^2 + \epsilon} \quad m = 1, \ldots, M,
$$

(3)

see Figure 2 for an example with $M = 3$. The updating scheme proposed in this paper explores the possibility of choosing between the different update equations based on comparison between the $M$ different error signals $e_m(k)$.

### 3. THE PROPOSED ALGORITHM

An adaptive linear filtering process can generally be divided in two parts the filtering (1) and the adaptation (2). In an echo cancellation environment, the filtering part generally is performed at every sample instant in order to produce a constant audio stream. Although it is most often efficient (in terms of convergence) to perform filter updating at every sample instant, it is not necessary. In practice, this might not even be possible due to complexity issues. This especially applies to acoustic echo cancellation environments where the dimension of the system filters is large.

One approach in a $M$-microphone system is to update only one adaptive filter every sample in a round-robin manner, that is, periodic NLMS. This also ensures equal (for all filters) and predictable convergence since the update occurrences are deterministic. The disadvantage is that convergence is slow.

This paper proposes another updating method which instead updates the filter with the largest output error. To illustrate the method, assume that $M = 3$ (3 adaptive filters), the present sample index is $k$, and filter 1 was updated at sample index $k - 1$, filter 3 at $k - 2$, and filter 2 at $k - 3$, as illustrated in Table 1. Thus, the available errors that can be used in the update at the present sample index $k$ are $e_1(k)$ for filter 1, $e_2(k)$, $e_3(k - 1)$ and $e_2(k - 2)$ for filter 2, and $e_3(k)$ and $e_3(k - 1)$ for filter 3. For example, the error $e_1(k - 2)$ cannot be used since it is related to the configuration of filter 1 prior to the latest update. From the available errors, the algorithm chooses the error with the largest magnitude and then performs the corresponding update (compare with (6) and (7) below).

#### Table 1: Example to illustrate the matrix $E(k)$.

| Sample index | Filter 1 | Filter 2 | Filter 3 |
|--------------|---------|---------|---------|
| $k$          | $e_1(k)$| $e_2(k)$| $e_3(k)$|
| $k - 1$      | Update  | $e_2(k - 1)$| $e_3(k - 1)$|
| $k - 2$      | X       | $e_2(k - 2)$| Update  |
| $k - 3$      | X       | Update  | X       |

An algorithm for the method is as follows. After filtering all $M$-output channels according to (1), the output errors from all filters are inserted in a $L \times M$ matrix

$$
E(k) = \begin{pmatrix}
    e_1(k) & e_2(k) & e_3(k) & \ldots & e_M(k)
\end{pmatrix}_{E(k - 1)},
$$

(4)

where $M$ is the number of adaptive filters (channels) and $L$ determines the number of previous samples to consider. The $L \times 1 \times M$ matrix $E(k - 1)$ consists of the $L - 1$ upper rows of $E(k - 1)$, that is,

$$
E(l + 1, m, k) = E(l, m, k - 1) \quad l = 1, \ldots, L - 1,
$$

$$
m = 1, \ldots, M,
$$

(5)

where $l$ and $m$ denote row and column indexes, respectively, and $E(l, m, k)$ is the element at row $l$ and column $m$ in $E(k)$.

The decision of which filter to update and with what output error (and corresponding input vector) is determined by the element in $E(k)$ with maximum absolute value,

$$
e_{\max}(k) = \max_{l m} |E(l, m, k)| \quad l = 1, \ldots, L,
$$

$$
m = 1, \ldots, M.
$$

(6)

The row and column indexes of the element in $E(k)$ with the maximum absolute value are denoted $l_{\max}(k)$ and $m_{\max}(k)$.
For clarity of presentation, the sample index is omitted, that is, \( l_{\text{max}} = l_{\text{max}}(k) \) and \( m_{\text{max}} = m_{\text{max}}(k) \).

The filter corresponding to the row index \( m_{\text{max}} \), that is, the filter \( \tilde{h}_{m_{\text{max}}} \), is then updated with

\[
\tilde{h}_{m_{\text{max}}}(k + 1) = \tilde{h}_{m_{\text{max}}}(k) + \frac{\mu e_{m_{\text{max}}}(k)x(k - l_{\text{max}} + 1)}{||x(k - l_{\text{max}} + 1)||^2 + \epsilon}.
\]

This filter update of filter \( \tilde{h}_{m_{\text{max}}} \) will make the error elements \( E(l, m_{\text{max}}, k) \), \( l = 1, \ldots, L \) obsolete, since these are errors generated by \( h_{m_{\text{max}}} \) prior to the update. Consequently, to avoid future erroneous updates, these elements should be set to 0, that is, set

\[
E(l, m_{\text{max}}, k) = 0 \quad \text{for} \quad l = 1, \ldots, L.
\]

An advantage over periodic NLMS is that the proposed structure does not limit the update to be based on the current input vector \( x(k) \), but allows updating based on previous input vectors as well, since the errors not yet used for an update are stored in \( E(k) \). Further, largest output-error update will concentrate the updates to the corresponding filter. This is normally a desired feature in an acoustic echo cancellation environment with multiple microphones. For example, consider the setup in Figure 1 with all adaptive filters fairly converged. If then one of the microphones is dislocated, this results in an echo-path change for the corresponding adaptive filter. Naturally, it is desired to concentrate all updates to this filter.

4. ANALYSIS

In the previously described scenario, where several input vectors are available but only one of them can be used for adaptive filter updating (due to complexity issues), it might seem intuitive to update with the input vector corresponding to the largest output error magnitude. In this section, it is shown analytically that, under certain assumptions, choosing the largest error maximizes the reduction.

The error deviation vector for the \( m \)th filter \( v_m(k) \) is defined as \( v_m(k) = h_m(k) - \tilde{h}_m(k) \), and the mean-squared deviation as \( D(k) = \mathbb{E} \{ ||v_m(k)||^2 \} \), where \( \mathbb{E} \{ \cdot \} \) denotes expectation [4]. Assume that no near-end sound is present, \( n(k) = 0 \), and no regularization is used, \( \epsilon = 0 \), and that the errors available for updating filter \( m \) are \( e_m(k - l_m) \) with \( l_m = 0, \ldots, L_m \) and \( L_m < L \), that is, the available errors in matrix \( E(k) \) that correspond to filter \( m \). Updating filter \( m \) using error \( e_m(k - l_m) \) gives

\[
||v_m(k + 1)||^2 = ||v_m(k) - \beta(k)e_m(k - l_m)x(k - l_m)||^2
\]

and by using

\[
e_m(k - l_m) = x(k - l_m)^T v_m(k) = v_m(k)^T x(k - l_m)
\]

in (9), the following is obtained:

\[
||v_m(k + 1)||^2 = ||v_m(k)\beta - (2\mu - \mu^2)||x(k - l_m)||^2 e_m^2(k - l_m).
\]

Thus, the difference in mean-square deviation from one sample to the next is given by

\[
D_m(k + 1) - D_m(k) = -2(2\mu - \mu^2)E\left\{ \frac{e_m^2(k - l_m)}{||x(k - l_m)||^2} \right\}
\]

which corresponds to a reduction under the assumption that \( 0 < \mu < 2 \).

Further, assuming small fluctuations in the input energy \( ||x(k)||^2 \) from one iteration to the next, that is, assuming

\[
||x(k)||^2 = ||x(k - 1)||^2 = \cdots = ||x(k - L_m + 1)||^2
\]

gives [4],

\[
D_m(k + 1) - D_m(k) = -2(2\mu - \mu^2)\frac{E\{e_m^2(k - l_m)\}}{E\{||x(k)||^2\}}
\]

The total reduction \( r(k) \) in deviation, considering all \( M \) filters is thus

\[
r(k) = \sum_{m=1}^{M} D_m(k + 1) - D_m(k).
\]

Only one filter is updated each time instant. Assume error \( E(l, m, k) \) is chosen for the update. Then \( r(k) \) is given by

\[
r(k) = -2(2\mu - \mu^2)\frac{E\{E^2(l, m, k)\}}{E\{||x(k)||^2\}}.
\]

From (16), it can be seen that the reduction is maximized if \( e_{\text{max}}(k) \), (see (16)), is chosen for the update, that is, as done in the proposed algorithm.

The proposed algorithm can be seen as a version of the periodic NLMS. Analysis of convergence, stability, and robustness for this branch of (N)LMS algorithms are provided in, for example, [5, 15].

5. COMPLEXITY AND IMPLEMENTATION

The algorithm proposed in this paper is aimed for implementation in a general digital signal processor (DSP), typically allowing multiply add and accumulate arithmetic operation to be performed in parallel with memory reads and/or writes (e.g., [16]). In such a processor, the filtering operation can be achieved in \( N \) instructions and the NLMS update will require \( 2N \) instructions. Both the filtering and the update require two memory reads, one addition and one multiplication per coefficient, which can be performed by the DSP in one instruction. However, the result from the filter update is not accumulated but it needs to be written back to memory. Therefore, the need for two instructions per coefficient for the update operation.
Suppose an $M$-channel system with the same number of adaptive filters, all with the length of $N$. The standard NLMS updating thus requires $3MN$ DSP instructions.

Updating the matrix $E(k)$, (4), can be implemented using circular buffering and thus requires only $M$ store instructions (possible pointer modifications disregarded), while clearing of $E(k)$, (8), takes a maximum of $L$ instructions (also disregarding possible pointer modifications). Searching for the maximum absolute valued element in $E(k)$, (6), requires a maximum of $2LM$ instructions ($LM$ abs-instructions and $LM$ max-instructions). The parameter $\|x(k)\|^2$ can be calculated very efficient through recursion, that is,

$$\|x(k)\|^2 = \|x(k - 1)\|^2 + x^2(k) - x^2(k - N),$$

and its computational complexity can be disregarded in this case.

All together, this means that the number of DSP instructions required for the proposed solution can be approximated with

$$MN + M + L + 2ML + 2N.$$

For acoustic echo cancellation, $N$ is generally quite large ($\gg 1000$) due to room reverberation time. In this case, we typically have $N \gg L$ and $N \gg M$, which means that (18) is approximately $N(M + 2)$. The complexity reduction in comparison with standard NLMS updating is then

$$\frac{M + 2}{3M},$$

which for $M = 3$ gives a complexity reduction of nearly a half ($5/9$). For higher values of $M$, the reduction is even larger. Further reduction in complexity can also be achieved if updates are performed say every other or every third sample.

6. SIMULATIONS

The performance of the proposed method was evaluated through simulations with speech as input signal. Three impulse responses ($h_1$, $h_2$, and $h_3$), shown in Figure 3, all of length $N = 1800$ were measured with three microphones, according to the constellation in Figure 1, in a normal office. The acoustic coupling between the loudspeaker and the closest microphone, AC1, was manually normalized to 0 dB and the coupling between the loudspeaker and the second and third microphones, AC2 and AC3, were then estimated to $-6$ dB and $-7$ dB, respectively. Thus, $10 \log_{10}(\|h_2\|^2/\|h_1\|^2) = -6$ dB and $10 \log_{10}(\|h_3\|^2/\|h_1\|^2) = -7$ dB.

Output signals $y_1(k)$, $y_2(k)$, and $y_3(k)$ were obtained by filtering the input signal $x(k)$ with the three obtained impulse responses and adding noise,

$$y_1(k) = x(k)^T h_1 + n_1(k),$$
$$y_2(k) = x(k)^T h_2 + n_2(k),$$
$$y_3(k) = x(k)^T h_3 + n_3(k).$$

The noise sources $n_1(k)$, $n_2(k)$, and $n_3(k)$ were independent, but had the same characteristics (bandlimited flat spectrum). Echo-to-noise ratio was approximately 40 dB for microphone 1 and 34 dB and 33 dB for microphones 2 and 3, respectively.

In the simulations four low-complexity methods of similar complexity were compared; the periodic (N)LMS [5], random NLMS (similar to SPU-LMS [10]) selecting which filter to be updated in a stochastic manner (with all filters having equal probability of an update), M-Max NLMS [6], and the proposed NLMS. The performance of the full update NLMS is also shown for comparison. The periodic NLMS, random NLMS, and the proposed method limit the updates to one whole filter at each time interval, while M-Max NLMS instead updates all filters but only does this for a subset (1/3 in this case) of all coefficients. However, since M-Max NLMS requires sorting of the input vectors, the complexity for this method is somewhat larger ($2 \log_2 N + 2$ comparisons and $N - 1/2$ memory transfers [9]). Zero initial coefficients were used for all filters and methods. The result is presented in Figure 4, where the normalized filter mismatch, calculated as

$$10 \log_{10} \left( \frac{\|h_m - \hat{h}_m(k)\|^2}{\|h_m\|^2} \right) m = 1, 2, 3,$$

for the three individual filters and solutions are presented. Of the four variants with similar complexity, the proposed method is clearly superior to the conventional periodic
Figure 4: Mismatch for the evaluated methods.

Figure 5: Mismatch for the evaluated methods, where an echo-path change occurs for filter 2 after 55 seconds.

NLMS and also to the random NLMS. The performance of the M-Max NLMS and the proposed solution is comparable, although the proposed solution performs better or equal for all filters.

The algorithm automatically concentrates computational resources to filters with large error signals. This is demonstrated in Figure 5, where filter 2 undergoes an echo-path change, that is, a dislocation of the microphone. In Figure 5, it can be seen that the proposed algorithm basically follows the curve of the full update NLMS immediately after the echo-path changes.

If one specific microphone is subject to an extreme acoustic situation, for example, it is placed in another room or placed immediately next to a strong noise source, there is a risk of “getting stuck,” that is, the corresponding filter has large output error for all input vectors and thus is updated all
the time. This problem can be reduced by setting a limit on the lowest rate of updates for a filter, that is, if filter \( m \) has not been updated for the last \( U \) samples it is forced to update the next iteration. However, this does not resolve the issue optimally. A more sophisticated method is to monitor the echo reduction of the filters and bypass or reduce the resources allocated to filters not providing significant error reduction. Implementing these extra functions will of course add complexity.

7. CONCLUSIONS

In an acoustic multichannel solution with multiple adaptive filters, the computation power required to update all filters every sample can be vast. This paper has presented a solution which updates only one filter every sample and thus significantly reduces the complexity, while still performing well in terms of convergence speed. The solution also handles echo-path changes well, since the most misadjusted filter gets the most computation power, which often is a desirable feature in practice.

ACKNOWLEDGMENT

The authors would like to thank the Swedish Knowledge Foundation (KKS) for funding.

REFERENCES

[1] E. Hansler and G. Schmidt, Acoustic Echo and Noise Control: A Practical Approach, John Wiley & Sons, New York, NY, USA, 2004.
[2] M. M. Sondhi, “An adaptive echo canceler,” Bell System Technical Journal, vol. 46, no. 3, pp. 497–510, 1967.
[3] B. Widrow and S. D. Stearns, Adaptive Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, USA, 1985.
[4] S. Haykin, Adaptive Filter Theory, Prentice-Hall, Englewood Cliffs, NJ, USA, 4th edition, 2002.
[5] S. C. Douglas, “Adaptive filters employing partial updates,” IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 44, no. 3, pp. 209–216, 1997.
[6] T. Aboulnasr and K. Mayyas, “Complexity reduction of the NLMS algorithm via selective coefficient update,” IEEE Transactions on Signal Processing, vol. 47, no. 5, pp. 1421–1424, 1999.
[7] P. A. Naylor and W. Sherliker, “A short-sort M-Max NLMS partial-update adaptive filter with applications to echo cancellation,” in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’03), vol. 3, pp. 373–376, Hong Kong, April 2003.
[8] K. Dogancay and O. Tanrikulu, “Adaptive filtering algorithms with selective partial updates,” IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 48, no. 8, pp. 762–769, 2001.
[9] T. Schertler, “Selective block update of NLMS type algorithms,” in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’98), vol. 3, pp. 1717–1720, Seattle, Wash, USA, May 1998.
[10] M. Godavarti and A. O. Hero III, “Partial update LMS algorithms,” IEEE Transactions on Signal Processing, vol. 53, no. 7, pp. 2382–2399, 2005.
Research Article

Time-Domain Convolutive Blind Source Separation Employing Selective-Tap Adaptive Algorithms

Qiongfeng Pan and Tyseer Aboulnasr

School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada K1N 6N5

Received 30 June 2006; Accepted 24 January 2007

Recommended by Patrick A. Naylor

We investigate novel algorithms to improve the convergence and reduce the complexity of time-domain convolutive blind source separation (BSS) algorithms. First, we propose MMax partial update time-domain convolutive BSS (MMax BSS) algorithm. We demonstrate that the partial update scheme applied in the MMax LMS algorithm for single channel can be extended to multichannel time-domain convolutive BSS with little deterioration in performance and possible computational complexity saving. Next, we propose an exclusive maximum selective-tap time-domain convolutive BSS algorithm (XM BSS) that reduces the interchannel coherence of the tap-input vectors and improves the conditioning of the autocorrelation matrix resulting in improved convergence rate and reduced misalignment. Moreover, the computational complexity is reduced since only half of the tap inputs are selected for updating. Simulation results have shown a significant improvement in convergence rate compared to existing techniques.

Copyright © 2007 Q. Pan and T. Aboulnasr. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Blind source separation (BSS) [1, 2] is an established area of work estimating source signals based on information about observed mixed signals at the sensors, that is, the estimation is performed without exploiting information about either the source signals or the mixing system. Independent component analysis (ICA) [3] is the main statistical tool for dealing with the BSS problem with the assumption that the source signals are mutually independent. In the instantaneous BSS case, signals are mixed instantaneously and ICA algorithms can be directly employed to separate the mixtures. However, in a realistic environment, signals are always mixed in convolutive manner because of propagation delay and reverberation effects. Therefore, much research deals with convolutive blind source separation based on extending instantaneous blind source separation or independent component analysis to convolutive case.

The straightforward choice in time-domain convolutive blind source separation is based on directly extending instantaneous BSS to the convolutive case [4, 5]. This natural approach achieves good separation results once the algorithm converges. However, time-domain convolutive blind source separation suffers from high computational complexity and low convergence rate, especially for systems requiring long FIR filters for the separation.

Frequency domain convolutive BSS [6, 7] was proposed to deal with the expensive computational complexity problem of time-domain BSS. In frequency domain BSS, complex-valued ICA for instantaneous BSS is employed in every frequency bin independently. The advantage of this approach is that any existing complex-valued instantaneous BSS algorithm can be used and the computational complexity is reduced by exploiting the FFT for the computation of convolution which is the basis of popularity of frequency domain approaches. However, the permutation and scaling ambiguity in the ICA algorithm, which is not a problem for instantaneous BSS, becomes a serious problem in frequency domain convolutive BSS. Since frequency domain convolutive BSS is performed by instantaneous BSS at each frequency bin separately, the order and the scale of the unmixed signals are random because of the inherent ambiguity of ICA algorithms. When we transform the separated signals back from frequency domain to time domain, the components at a given frequency bin may not come from the same source signal and may not have a consistent scale factor. Thus, we need to align these components and adjust the scale in each frequency bin so that a separated signal in time domain is obtained from frequency components of the same source signal and with consistent amplitude. This is well known as the
permutation and scaling problem of frequency domain convolutive BSS [8, 9]. These built-in problems in frequency domain approaches make it worthwhile to reconsider ways of reducing the complexity of time-domain approaches and improving their convergence rates.

In recent years, several partial update adaptive algorithms were proposed to model single-channel systems with reduced overall system complexity by updating only a subset of coefficients. Within these partial update algorithms, the MMax NLMS in [10] was reported to have the closest performance to the full update case for any given number of coefficients to be updated. In [11], the MMax selective-tap strategy was extended to the two-channel case to exclusively select coefficients corresponding to the maximum inputs as a means to reduce interchannel coherence in stereophonic acoustic echo cancellation rather than as a way to reduce complexity. Simulation results for this exclusive maximum adaptive algorithm show that it can significantly improve the convergence rate compared with existing stereophonic echo cancellation techniques.

In this paper, we propose using these reduced complexity approaches in time-domain BSS to address complexity and low convergence problems. First, we propose MMax natural gradient-based partial update time-domain convolutive BSS algorithm (MMax BSS). In this algorithm, only a subset of coefficients in the separation system gets updated at every iteration. We demonstrate that the partial update scheme applied in the MMax LMS algorithm for a single channel can be extended to the multichannel time-domain convolutive BSS with little deterioration in performance and possible computational complexity saving. By employing selective-tap strategies used for stereophonic acoustic echo cancellation [11], we propose exclusive maximum selective-tap time-domain convolutive BSS algorithm (XM BSS). The exclusive tap-selection update procedure reduces the interchannel coherence of the tap-input vectors and improves the conditioning of the autocorrelation matrix so as to accelerate convergence rate and reduce the misalignment. The computational complexity is reduced as well since only half of the tap inputs are selected for updating (note that some overhead is needed to select the set to be updated). Simulation results have shown a significant improvement in convergence rate compared with existing techniques. As far as we know, the application of partial update and selective-tap update schemes to time-domain BSS algorithm is in itself novel.

BSS algorithms are generally preceded by a prewhitening stage that aims to reduce the correlation between the different input sources (as opposed to regular whitening where correlation between different samples of the same source is reduced). This decorrelation step leads to a subsequent separation matrix that is orthogonal and less ill-conditioned. The proposed partial update BSS algorithm incorporates this whitening concept into the separation process by adaptively reducing the interchannel coherence of the tap-input vectors.

The rest of this paper is organized as follows. In Section 2, we review blind source separation and its challenges in time domain and frequency domain. In Section 3, we review the single-channel MMax partial update adaptive algorithm for linear filters. In Section 4, we review exclusive maximum selective-tap adaptive algorithm for stereophonic echo cancellation. We propose the MMax partial update time-domain convolutive BSS algorithm in Section 5 and the exclusive maximum update time-domain convolutive BSS algorithm in Section 6. The tools for assessing the quality of the separation are presented in Section 7 and simulation results for the proposed algorithms for generated gamma signals and speech signals are presented in Section 8. In Section 9, we draw our conclusions from our work.

2. BLIND SOURCE SEPARATION

2.1. Instantaneous time-domain BSS

Blind source separation (BSS) is a very versatile tool for signal separation in a number of applications utilizing observed mixtures and the independence assumption. For instantaneous mixtures, independent component analysis (ICA) can be employed directly to separate the mixed signals.

The ICA-based algorithm for instantaneous blind source separation requires the output signals to be as independent as possible. Different algorithms can be obtained based on how this independence is measured. The instantaneous time-domain BSS structure is shown in Figure 1. In this paper, we use the Kullback-Leibler divergence to measure independence and obtain the BSS algorithm as follows:

\[
\begin{align*}
\textbf{x} &= \textbf{As}, \\
\textbf{y} &= \textbf{Wx},
\end{align*}
\]

where \( \textbf{s} = [s_1, \ldots, s_N]^T \) is the vector of source signals, \( \textbf{x} = [x_1, \ldots, x_M]^T \) is the vector of mixture signals, \( \textbf{y} = [y_1, \ldots, y_N]^T \) is the vector of separated signals, \( \textbf{A} \) and \( \textbf{W} \) are instantaneous mixing and unmixing systems and can be described as

\[
\textbf{A} = \begin{bmatrix}
a_{11} & \cdots & a_{1N} \\
\vdots & \ddots & \vdots \\
da_{M1} & \cdots & a_{MN}
\end{bmatrix}, \quad \textbf{W} = \begin{bmatrix}
w_{11} & \cdots & w_{1M} \\
\vdots & \ddots & \vdots \\
w_{N1} & \cdots & w_{NM}
\end{bmatrix}.
\]

The Kullback-Leibler divergence of the output signal vector
where $D(p(y) \mid q(y))$ is the probability density of output signals, $p_i(y_i)$ is the probability density of output signal $y_i$, $q(y)$ is the joint probability density of output signals:

$$D(p(y) \mid q(y)) = \int p(y) \log \frac{p(y)}{\prod_{i=1}^{N} p_i(y_i)} \, dy,$$

(3)

where $p(y)$ is the probability density of output signals, $p_i(y_i)$ is the probability density of output signal $y_i$, $q(y)$ is the joint probability density of output signals:

$$D(p(y) \mid q(y)) = \int p(y) \log p(y) - \sum_{i=1}^{N} \left( \int p(y) \log p_i(y_i) \right)$$

$$= -H(y) + \sum_{i=1}^{N} H_i(y_i)$$

$$= -H(x) - \log |\text{det}(W)| - \sum_{i=1}^{N} E[\log (p_i(y_i))],$$

(4)

where $H(\cdot)$ is the entropy operation.

Using standard gradient

$$\Delta D = \frac{\partial D}{\partial W} = -\frac{\partial}{\partial W} H(x) - \frac{\partial}{\partial W} \log (|\text{det}(W)|)$$

$$+ \frac{\partial}{\partial W} \sum_{i=1}^{N} E[\log (p_i(y_i))]$$

$$= 0 - W^{-T} + E[\varphi(y)x^T],$$

(5)

where $\varphi(y) = [\partial p_1(y_1)/\partial y_1/p_1(y_1), \ldots, \partial p_N(y_N)/\partial y_N/p_N(y_N)]$ is a nonlinear function related to the probability density function of source signals, the coefficients $W$ in the unmixing system are then updated as follows:

$$W(k+1) = W(k) + \Delta W,$$

$$\Delta W_{\text{standard_grad}} = -\mu \frac{\partial D}{\partial W} = \mu(W^{-T} - E[\varphi(y)x^T]).$$

(6)

However, BSS algorithms have traditionally used the natural gradient [4] which is acknowledged as having better performance. In this case, $\Delta W$ is given by

$$\Delta W_{\text{natural_grad}} = -\mu \frac{\partial D}{\partial W} W^T W = \mu[I - E(\varphi(y)y^T)] W.$$

(7)

### 2.2. Convolutive BSS algorithm

The convolutive BSS model is illustrated in Figure 2. $N$ source signals $\{s_i(k)\}$, $1 \leq i \leq N$, pass through an unknown $N$-input, $M$-output linear time-invariant mixing system to yield the $M$ mixed signals $\{x_j(k)\}$. All source signals $s_i(k)$ are assumed to be statistically independent.

Defining the vectors $s(k) = [s_1(k) \cdots s_N(k)]^T$ and $x(k) = [x_1(k) \cdots x_M(k)]^T$, the mixing system can be represented as

$$[x_1(k) \cdots x_M(k)] = [h_{11}(l) \cdots h_{1N}(l)] \ast [s_1(k) \cdots s_N(k)],$$

(8)

where $\ast$ is convolution operation.

The $j$th sensor signal can be obtained by

$$x_j(k) = \sum_{i=1}^{N} h_{ji}(l)s_i(k-l),$$

(9)

where $h_{ji}(l)$ is the impulse response from source $i$ to sensor $j$. $L$ defines the order of the FIR filters used to model this impulse response.

The task of the convolutive BSS algorithm is to obtain an unmixing system such that the outputs of this system $y(k) = [y_1(k) \cdots y_N(k)]^T$ become mutually independent as the estimates of the $N$ source signals. The separation system typically consists of a set of FIR filters $w_{ij}(l)$ of length $Q$ each. The unmixing system can also be represented as

$$[y_1(k) \cdots y_N(k)] = [w_{11}(l) \cdots w_{1M}(l)] \ast [x_1(k) \cdots x_M(k)],$$

(10)

The $i$th output of the unmixing system is given as

$$y_i(k) = \sum_{j=1}^{M} w_{ij}(l)x_j(k-l).$$

(11)

By extending the instantaneous BSS algorithm to the convolutive case, we get the time-domain convolutive BSS algorithm as

$$\Delta W = -\mu \frac{\partial D}{\partial W} W^TW = \mu[I - E(\varphi(y)y^T)] W.$$ 

(12)

where $W$ the unmixing matrix with FIR filters as its components.

This approach is the natural extension and achieves good separation results once the algorithm converges. However, time-domain convolutive blind source separation suffers from high computational complexity and slow convergence rate, especially for systems with long FIR filters.

Convolutional BSS can also be performed in frequency domain by using short-time Fourier transform. This method is very popular for convolutive mixtures and is based on transforming the convolutive blind source separation problem into instantaneous BSS problem at every frequency bin.
The advantage of frequency domain convolutive BSS lies in three factors. First the computational complexity is reduced since the convolution operations are transferred into multiplication operations by short-time FFT. Second, the separation process can be performed in parallel at all frequency bins. Finally any complex-valued instantaneous ICA algorithm can be employed to deal with the separation at each frequency bin. However, the permutation and scaling algorithm can be employed to deal with the separation at each frequency bin. As a result, the order and the scale in each frequency bin so that a separated signal in the frequency domain convolutive BSS is performed by instantaneous BSS, which is not a problem for instantaneous BSS, becomes a serious problem in frequency domain convolutive BSS.

This problem can be illustrated by Figure 3. Frequency domain convolutive BSS is performed by instantaneous BSS at each frequency bin separately. As a result, the order and the scale of the unmixed signals are random because of the inherent indeterminacy of ICA algorithms. When we transform the separated signals back from frequency domain to time domain, the components at different frequency bins may not come from the same source signal and may not have consistent scale. Thus, we need to align the permutation and adjust the scale in each frequency bin so that a separated signal in time domain is obtained from frequency components of the same source signal and with consistent amplitude. This is not a simple problem.

3. PARTIAL UPDATE ADAPTIVE ALGORITHM

The basic idea of partial update adaptive filtering is to allow for the use of filters with a number of coefficients $L$ large enough to model the unknown system while reducing the overall complexity by updating only $M$ coefficients at a time. This results in considerable savings for $M \ll L$. Invariably, there are penalties for this partial update, the most obvious of which is reduced convergence rate. The question then becomes which coefficients should we update and how do we minimize the impact of the partial update on the overall filter performance. In this section, we review the MMax partial update adaptive algorithm for linear filters [10] since it forms the basis of our proposed MMax time-domain convolutive BSS algorithm.

Consider a standard adaptive filter set-up where $x(n)$ is the input, $y(n)$ is the output, and $d(n)$ is the desired output, all at instant $n$. The output error $e(n)$ is given by

$$e(n) = d(n) - y(n) = d(n) - w^T(n)x(n),$$

where $w(n)$ is the $L \times 1$ column vector of the filter coefficients and $x(n)$ is the $L \times 1$ column vector $x(n) = [x(n), \ldots, x(n-i), \ldots, x(n-L+1)]$ of the current and past inputs to the filter, both at instant $n$. The $i$th element of $w(n)$ is $w_i(n)$ and it multiplies the $i$th delayed input $x(n)$, $i = 0, \ldots, L - 1$.

The basic NLMS algorithm is known for its extreme simplicity provided for coefficient update as given by

$$w(n + 1) = w(n) + \mu\epsilon(n)x(n)||x(n)||^2,$$

where $\mu$ is the step size determining the speed of convergence and the steady state error.

In the single-channel MMax NLMS algorithm [10], for an adaptive filter of length $L$, the set of $M$ coefficients to be updated is selected as the one that provides the maximum reduction in error. It is shown in [10] that this criterion reduces to the set of coefficients multiplying inputs $x(n-i)$ with the largest magnitude using the standard NLMS update equation. This selective-tap updating can be expressed as

$$w(n + 1) = w(n) + \mu Q(n)\epsilon(n)x(n)||x(n)||^2,$$

where $Q(n)$ is the tap–selection matrix as

$$Q(n) = \text{diag}\{q(n)\},$$

$$q_i(n) = \begin{cases} 1, & |x(n-i-1)| \in \{M \text{ maxima of } |x(n)|\} \\ 0, & \text{otherwise.} \end{cases}$$

An analysis of the mean square error convergence is provided in [10] based on matrix formulation of data-dependent partial updates. Based on the analysis, it was shown that the MMax algorithm provides the closest performance to the full update case for any given number of coefficients to be updated. This was also confirmed in [12].

4. EXCLUSIVE MAXIMUM SELECTIVE-TAP ADAPTIVE ALGORITHM

Recently, an exclusive maximum (XM) partial update algorithm was proposed in [11] to deal with stereophonic echo cancellation. The XM algorithm was motivated by MMax partial update scheme [10] as both select a subset of coefficients for updating in every adaptive iteration. However, in the XM partial update, the goal is not to reduce computational complexity. Rather the exclusive maximum tap–selection strategy was proposed to reduce interchannel coherence in a two-channel stereo system and improve the conditioning of the input vector autocorrelation matrix. We now review the algorithm in [11] here since it forms the basis of our proposed XM time-domain convolutive BSS algorithm.

In stereophonic acoustic environment, the stereophonic signals $x_1(n)$ and $x_2(n)$ are transmitted to louder speakers in the receiving room and coupled to the microphones in this room by the room impulse responses. In stereophonic acoustic echo cancellation, these coupled acoustic echoes have to be cancelled. Let the receiving room impulse responses for
\( x_1(n) \) and \( x_2(n) \) be \( h_1(n) \) and \( h_2(n) \), respectively. Two adaptive filters \( \hat{h}_1(n) \) and \( \hat{h}_2(n) \) of length \( L \) in stereophonic acoustic echo canceller are updated to estimate \( h_1(n) \) and \( h_2(n) \). The desired signal for the adaptive filters is

\[
d(n) = \sum_{j=1}^{2} \hat{h}_j^T(n)x_j(n),
\]

where \( h_j(n) = [h_{j0}(n), h_{j1}(n), \ldots, h_{j,L-1}(n)]^T \) and \( x_j(n) = [x_{j1}(n), x_{j2}(n-1), \ldots, x_{j1}(n-L+1)]^T \).

Thus, the error signal is

\[
e(n) = d(n) - \sum_{j=1}^{2} \hat{h}_j^T(n)x_j(n).
\]

Adaptive algorithms such as LMS, NLMS, RLS, and affine projection (AP) can be used to update these two adaptive filters \( \hat{h}_1(n) \) and \( \hat{h}_2(n) \). The exclusive maximum tap-selection scheme is outlined in the following.

1. At each iteration, calculate the interchannel tap magnitude difference vector as \( \mathbf{p} = |x_1| - |x_2| \).
2. Sort \( \mathbf{p} \) in descending order as \( \mathbf{p} = [p_1, \ldots, p_L]^T \), \( p_1 > p_2 > \cdots > p_L \).
3. Order \( x_1 \) and \( x_2 \) according to the sorting of \( \mathbf{p} \) as \( \tilde{x}_1 = [\tilde{x}_1(n), \tilde{x}_1(n-1), \ldots, \tilde{x}_1(n-L+1)]^T \) and \( \tilde{x}_2 = [\tilde{x}_2(n), \tilde{x}_2(n-1), \ldots, \tilde{x}_2(n-L+1)]^T \).
4. The first channel coefficients corresponding to the \( M \) largest elements of \( \mathbf{p} \) get updated and the second channel coefficients corresponding to \( M \) smallest elements of \( \mathbf{p} \) get updated.

It was shown in [11] that this update mechanism applying to LMS, NLMS, RLS, and affine projection (AP) algorithms results in significantly better convergence rate than their existing corresponding algorithms.

5. PROPOSED MMAX PARTIAL UPDATE TIME-DOMAIN CONVOLUTIVE BSS ALGORITHM

From the description of MMax partial update in Section 3, we know that the principle of MMax partial update algorithm for single channel is to update the subset of coefficients which has the most impact on \( \Delta \mathbf{w} \). Our proposed MMax partial update convolutive BSS algorithm is based on the same principle.

In the MMax LMS algorithm [10], given \( \Delta \mathbf{w}(n) = e(n)x(n) \), the \( e(n) \) is common to all elements of \( \Delta \mathbf{w}(n) \), then the larger the \( |x(n-i)| \), the larger its impact on error. Thus, in MMax LMS algorithm, the coefficients corresponding to \( M \) largest values in \( |x(n)| \) are updated.

However, in time-domain convolutive BSS, \( \Delta \mathbf{W} \) is as follows:

\[
\Delta \mathbf{W} = -\mu \frac{\partial D}{\partial \mathbf{W}} \mathbf{W}^T \mathbf{W} = \mu [\mathbf{I} - E(\phi(y)y^T)] \mathbf{W}.
\]

Every element of \( \mathbf{W} \) is an FIR filter and there is no common value for all elements of \( \Delta \mathbf{W} \). Based on MMax partial update principle, the coefficients with the \( M \) largest values of \( \Delta \mathbf{W}_{ij} \) are the ones to be updated. We show this algorithm using a 2-by-2 system as an example in Algorithm 1.

From the algorithm description, the challenge compared to the MMax LMS algorithm [10] is that we need to sort the elements in \( \Delta \mathbf{W}_{ij} \) in every iteration, as opposed to simply identifying the location of one new sample in an already ordered set. However, we only need to update the selected subset of coefficients, which results in some savings.

6. PROPOSED EXCLUSIVE MAXIMUM SELECTIVE-TAP TIME-DOMAIN CONVOLUTIVE BSS ALGORITHM

As we already know from Section 4, exclusive maximum tap selection can reduce interchannel correlation and improve the conditioning of the input autocorrelation matrix. In this section, we examine the effect of tap selection on interchannel coherence reduction and extend this idea to our multi-channel blind source separation case.

6.1. Interchannel decorrelation by tap selection

The squared coherence function of \( x_1, x_2 \) is defined as

\[
C_{x_1x_2}(f) = \frac{|P_{x_1x_2}(f)|^2}{P_{x_1x_1}(f)P_{x_2x_2}(f)},
\]

where \( P_{x_1x_2}(f) \) is the cross-power spectrum between the two mixtures \( x_1, x_2 \) and \( f \) is the normalized frequency [11].
A two-input two-output system is considered in this section. The mixing system used in the simulation is as follows:

\[
H = \begin{bmatrix}
h_{11} & h_{12} \\
h_{21} & h_{22}
\end{bmatrix},
\]

\[
h_{11} = \begin{bmatrix}
1 & 0.8 & -0.2 & 0.78 & 0.4 & -0.2 & 0.1
\end{bmatrix},
\]

\[
h_{22} = \begin{bmatrix}
0.8 & 0.6 & 0.1 & -0.1 & 0.3 & -0.2 & 0.1
\end{bmatrix},
\]

(21)

where \( b \) is an independent white Gaussian noise with zero mean.

In the simulation, we set \( \gamma = 0.9 \) to reflect the high interchannel correlation found in practice between the observed mixtures in a convolutive environment. The two-tap input signals \( s_1 \) and \( s_2 \) are generated as zero mean, unit variance gamma signals. The mixtures \( x_1 \) and \( x_2 \) are obtained from the following equations:

\[
x_1 = s_1 * h_{11} + s_2 * h_{12},
\]

\[
x_2 = s_1 * h_{21} + s_2 * h_{22},
\]

(22)

where \( * \) is convolution operation.

The squared coherence for the \( x_1 \) and \( x_2 \) with full taps selected is shown in Figure 4. In Figure 5, the squared coherence for inputs with taps selected according to the MMax selection criterion as described in Section 4 is shown. We can see that the correlation is reduced, but not significantly. Figure 6 shows the squared coherence for signals with exclusive tap selected, that is, the selection of the same tap index in both channels is not permitted. We can see that the correlation is reduced significantly. This confirms that exclusive tap-selection strategy does indeed reduce interchannel coherence and as such improves the conditioning of the input autocorrelation matrix even in the mixing environment of blind source separation case.

6.2. Proposed XM update algorithm for time-domain convolutive BSS

As a result of improved conditioning of input autocorrelation matrix, we expect improved convergence rate in time-domain convolutive BSS when using this update algorithm for a two-by-two blind source separation system.

Based on the exclusive maximum tap-selection scheme proposed in [11], we propose the exclusive maximum time-domain convolutive BSS algorithm (XM BSS) as follows.

Define \( p \) as the interchannel tap input magnitude difference vector at time \( n \) as

\[
p = |x_1| - |x_2|.
\]

(23)
Sort \( \mathbf{p} \) in descending order as

\[
\tilde{\mathbf{p}} = [\tilde{p}_1, \ldots, \tilde{p}_L]^T, \quad \tilde{p}_1 > \tilde{p}_2 > \cdots > \tilde{p}_L.
\] (24)

Order \( \mathbf{x}_1 \) and \( \mathbf{x}_2 \) according to the sorting of \( \tilde{\mathbf{p}} \) such that \( \tilde{x}_1(n-i) \) and \( \tilde{x}_2(n-i) \) correspond to \( \tilde{p}_i = |x_1(n-i)| - |x_2(n-i)| \).

Taps corresponding to the \( M = 0.5L \) largest elements of the input magnitude difference vector \( \mathbf{p} \) in the first channel and the \( M \) smallest elements of \( \mathbf{p} \) in the second channel are selected for the updating of the output signal \( y_1 \); Taps corresponding to the \( M = 0.5L \) largest elements of the input magnitude difference vector \( \mathbf{p} \) in the second channel and the \( M \) smallest elements of \( \mathbf{p} \) in the first channel are selected for the updating of the output signal \( y_2 \). The detailed algorithm is shown in Algorithm 2.

6.3. Computational complexity of the proposed algorithm

The complexity is defined as the total number of multiplications and comparisons per sample period for each channel. In XM convolutive BSS algorithm, we need to sort the interchannel tap input magnitude difference vector. For an unmixing system with filter length \( L \), we require at most \( 2+2\log_2 L \) comparisons per sample period by the SORTLINE procedure [13]. However, the number of multiplications required for computing convolution per sample period is reduced from \( 4L \) to \( 2L \) for a two-by-two BSS system. Thus, the overall computational complexity is still reduced provided \( L > 2 \), which is always satisfied for convolutive BSS case.

7. SEPARATION PERFORMANCE EVALUATION

In this section, we describe separation performance evaluation measurement used in our simulations.

7.1. Performance evaluation by signal-to-interference ratio

The performance of blind source separation systems can be evaluated by the signal-to-interference ratio (SIR) which is defined as the power ratio between the target component and the interference components [14].

In basic instantaneous BSS model, the mixing system is represented with \( \mathbf{A} \), the unmixing system is represented with \( \mathbf{W} \), the global system can be presented as \( \mathbf{P} = \mathbf{A} \star \mathbf{H} \). Each element in \( i \)th row and \( j \)th column of \( \mathbf{P} \) is a scalar \( p_{ij} \). The SIR of output \( i \) is obtained as

\[
\text{SIR}_i = 10 \log_{10} \frac{E[|p_{ii}\hat{s}_i|^2]}{E[\sum_{j \neq i} p_{ij}\hat{s}_j]^2}] \text{dB}
\] (25)

for instantaneous BSS case.

In the convolutive BSS model, the mixing system is represented with \( \mathbf{H} \), the unmixing system with \( \mathbf{W} \). We can express the global system as \( \mathbf{P} = \mathbf{W} \star \mathbf{H} \) and each element in \( \mathbf{P} \) is a vector \( \mathbf{p}_{ij} \).

\[
(1) \text{Initialize } \mathbf{W} = \begin{bmatrix} \mathbf{w}_{11} & \mathbf{w}_{12} \\ \mathbf{w}_{21} & \mathbf{w}_{22} \end{bmatrix}
\]

(2) Iteration \( k \)

\[
\begin{align*}
\mathbf{x}_1 &= \{x_1(k), x_1(k-1), \ldots, x_1(k-L+1)\}; \\
\mathbf{x}_2 &= \{x_2(k), x_2(k-1), \ldots, x_2(k-L+1)\}; \\
\mathbf{p} &= |\mathbf{x}_1| - |\mathbf{x}_2|; \\
\tilde{\mathbf{x}}_{11} &= \mathbf{Q}_{11} \times \tilde{\mathbf{x}}_1; \quad \tilde{\mathbf{x}}_{21} = \mathbf{Q}_{21} \times \tilde{\mathbf{x}}_1; \\
\tilde{\mathbf{x}}_{12} &= \mathbf{Q}_{12} \times \tilde{\mathbf{x}}_2; \quad \tilde{\mathbf{x}}_{22} = \mathbf{Q}_{22} \times \tilde{\mathbf{x}}_2; \\
\mathbf{Q}_{11} &= \text{diag} [\mathbf{q}_{11}]; \\
\mathbf{Q}_{12} &= \text{diag} [\mathbf{q}_{12}]; \\
\mathbf{q}_{11}(m) &= \begin{cases} 1 & p(m) \in |M \text{ maxima of } \mathbf{p}| \\
0 & \text{otherwise}; \end{cases} \\
\mathbf{q}_{12}(m) &= \begin{cases} 1 & p(m) \in |M \text{ maxima of } \mathbf{p}| \\
0 & \text{otherwise}; \end{cases} \\
\mathbf{Q}_{21} &= \text{diag} [\mathbf{q}_{21}]; \\
\mathbf{Q}_{22} &= \text{diag} [\mathbf{q}_{22}]; \\
\mathbf{q}_{21}(m) &= \begin{cases} 1 & p(m) \in |M \text{ minima of } \mathbf{p}| \\
0 & \text{otherwise}; \end{cases} \\
\mathbf{q}_{22}(m) &= \begin{cases} 1 & p(m) \in |M \text{ minima of } \mathbf{p}| \\
0 & \text{otherwise}; \end{cases} \end{align*}
\]

\[
\mathbf{y}_1 = \mathbf{w}_{11} \times \tilde{\mathbf{x}}_{11} + \mathbf{w}_{12} \times \tilde{\mathbf{x}}_{12}; \\
\mathbf{y}_2 = \mathbf{w}_{21} \times \tilde{\mathbf{x}}_{21} + \mathbf{w}_{22} \times \tilde{\mathbf{x}}_{22}; \\
u_1 &= \tanh (\mathbf{y}_1); \\
u_2 &= \tanh (\mathbf{y}_2); \\
\Delta \mathbf{W} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \times \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}; \\
\mathbf{W} &= \mathbf{W} + \mu \times \Delta \mathbf{W}; \\
k &= k + 1.
\]

(3) Go to 2 to start another iteration.

(4) Calculate separated signals as

\[
\begin{align*}
y_1 &= \mathbf{w}_{11} \times \tilde{x}_1^1 + \mathbf{w}_{12} \times \tilde{x}_1^2; \\
y_2 &= \mathbf{w}_{21} \times \tilde{x}_2^1 + \mathbf{w}_{22} \times \tilde{x}_2^2.
\end{align*}
\]

\textbf{Algorithm 2: XM convolutive BSS algorithm.}

The SIR of output \( i \) is obtained as

\[
\text{SIR}_i = 10 \log_{10} \frac{E[|p_{ii}\hat{s}_i|^2]}{E[\sum_{j \neq i} p_{ij}\hat{s}_j]^2}] \text{dB}
\] (26)

for convolutive BSS case, where \( \ast \) is the convolution operation and \( E\{\} \) is the expectation operation.
7.2. Performance evaluation by PESQ

When the target signal in our simulations is a speech signal, we will also use PESQ (perceptual evaluation of speech quality) as a measure confirming the quality of the separated signal. The PESQ standard [15] is described in the ITU-T P862 as a perceptual evaluation tool of speech quality. The key feature of the PESQ standard is that it uses a perceptual model analogous to the assessment by the human auditory system. The output of the PESQ is a measure of the subjective assessment quality of the degraded signal and is rated as a value between $-0.5$ and $4.5$ which is known as the mean opinion score (MOS). The larger the score, the better the speech quality.

8. SIMULATIONS

8.1. Experiment setup

In the following simulations, our source signals $s_1$ and $s_2$ are generated as gamma signals or speech signals. The gamma signals are generated with zero mean, unit variance. The speech signals used in our simulations include 3 female speeches and 3 male speeches with sample rate 8000 Hz to form 9 combinations. A simple mixing system is used in our simulations to demonstrate and compare separation performance.

The mixing system is given by

$$H = \begin{bmatrix} 1.0 & 0.2 & 0.4 & 0.7 \\ 0.0 & 1.0 & 0.0 & 0.2 \\ -0.75 & 0.5 & -0.3 & 0.2 \end{bmatrix}. \quad (27)$$

The mixture signals are obtained by convolving the source signals with the mixing system. The filter length in the separation system is set at 64.

In the following, we will compare the separation performance of the regular convolutive BSS algorithm, MMax partial update BSS algorithm, and XM selective-tap BSS algorithm.

8.2. MMax partial update time-domain BSS algorithm for convolutive mixture

In this simulation, we test the performance of MMax partial update time-domain BSS algorithm for convolutive mixtures. In the following diagram, “reg” means regular time-domain BSS algorithm; “par56” means MMax partial update time domain BSS algorithm with $M = 56$; “par48” means MMax partial update time-domain BSS algorithm with $M = 48$; “par32” means MMax partial update time-domain BSS algorithm with $M = 32$, where $M$ is the number of coefficients updated at each iteration in a given channel.

In the first experiment, we use generated gamma signals as the original signals and use (9) to get the mixture signals. The performance of regular time-domain convolutive BSS algorithm and MMax partial update convolutive BSS algorithm evaluated by the SIR measure defined in (26) is shown in Figures 7 and 8.

From these diagrams, we can see that as expected, the MMax partial update convolutive BSS and MMax partial update BSS for gamma signal measured by SIR for the first output.

In the second experiment, we use speech signals as the original signals and use the same mixing system to get the mixture signals. In Figures 9 and 10, we show the performance of regular time-domain convolutive BSS algorithm and MMax partial update BSS convolutive algorithm for one
combination of speech signals, the separation performance is evaluated by SIR. The performance for other combinations of speech signals is similar to that shown in Figures 9 and 10.

Since we used speech signals in the second experiment, we also use PESQ to evaluate the separation performance. In the following, we evaluate the similarity between the mixtures, the separated signals from regular and MMax BSS algorithms with the original source signals by PESQ score. Table 1 shows the average PESQ evaluation results for different combinations of female and male speech signals, where (S1, S2) present the original source signals; (mix1, mix2) present the mixture signals; (regular out1, regular out2) present separated signals from regular BSS algorithm; (partial M = 56 out1, partial M = 56 out2) present separated signals from MMax BSS algorithm with M = 56; (partial M = 48 out1, partial M = 48 out2) present separated signals from MMax BSS algorithm with M = 48; (partial M = 32 out1, partial M = 32 out2) present separated signals from MMax BSS algorithm with M = 32.

From Table 1, we can see that the separation performance evaluated by PESQ is consistent with the SIR results. The separation algorithms make the separated signals more biased to one source signal and away from the other source signal. The separation performance evaluated by PESQ and SIR is also consistent with our informal listening tests.

From the above simulation results, we can see that similar to MMax NLMS algorithm for single-channel linear filters, there is a slight deterioration in performance of the proposed MMax partial update time-domain convolutive BSS algorithm as the number of updated coefficients is reduced. However, the performance at 50% coefficients updated is still quite acceptable.

8.3. Time-domain exclusive maximum selective-tap BSS for convolutive mixture

In this simulation, we test the performance of XM selective tap time-domain BSS algorithm for convolutive mixtures.

In the first experiment, we use generated gamma signals as the original signals and use (9) to get the mixture signals. The performance of regular time-domain convolutive BSS algorithm and XM selective-tap convolutive BSS algorithm evaluated by SIR is shown in Figures 11 and 12.

From Figures 11 and 12, we can see that XM BSS algorithm has much better convergence rate compared with regular BSS algorithm for generated gamma signals.

In the second experiment, we use speech signals as the original signals and use the same mixing system to get the mixture signals. In Figures 13 and 14, we show the performance of regular time-domain convolutive BSS algorithm and XM selective tap BSS convolutive algorithm for one combination of speech signals, the separation performance is evaluated by SIR. The performance for other combinations of speech signals is similar with that shown in Figures 13 and 14.

From the plots, we can see that the XM BSS algorithm has much better convergence rate compared with the regular BSS algorithm for both generated gamma signals and speech signals.

Since we used speech signals in the second experiment, we also use PESQ to evaluate the separation performance. In the following, we evaluate the similarity between the mixtures, the separated signals from regular and XM BSS algorithms with the original source signals by PESQ score. Table 2 shows the average PESQ evaluation results for different combinations of female and male speech signals, where (S1, S2) present the original source signals; (mix1, mix2) present the mixture signals; (regular BSS out1, out2) present separated...
Table 1: Average PESQ scores for mixtures and separated signals from regular BSS algorithm and MMax BSS algorithm.

| PESQ | Mixture | Regular | Partial M = 56 | Partial M = 48 | Partial M = 32 |
|------|---------|---------|---------------|---------------|---------------|
|      | mix1    | mix2    | out1          | out2          | out1          | out2          | out1          | out2          |
| S1   | 2.119   | 0.981   | 2.379         | 0.612         | 2.365         | 0.611         | 2.352         | 0.602         | 2.340         | 0.599         |
| S2   | 1.364   | 2.374   | 1.076         | 2.771         | 1.105         | 2.702         | 1.148         | 2.659         | 1.029         | 2.624         |

Figure 11: Separation performance of time-domain regular convolutive BSS and XM selective tap BSS for gamma signal measured by SIR for the first output.

Figure 12: Separation performance of time-domain regular convolutive BSS and XM selective tap BSS for gamma signal measured by SIR for the second output.

Figure 13: Separation performance of time-domain regular convolutive BSS and XM selective tap BSS for speech signal measured by SIR for the first output.

Figure 14: Separation performance of time-domain regular convolutive BSS and XM selective tap BSS for speech signal measured by SIR for the second output.
Table 2: Average PESQ scores for mixtures and separated signals from regular BSS algorithm and XM BSS algorithm.

| PESQ | Mixture | Regular BSS | Xmax BSS |
|------|---------|-------------|----------|
|      | mix1    | out1        | out1     |
|      | mix2    | out2        | out2     |
| S1   | 1.871   | 0.948       | 2.037    |
|      |         | 0.591       | 1.055    |
|      |         | 2.560       | 2.643    |
|      |         | 0.463       | 1.215    |
| S2   | 1.583   | 2.255       | 1.215    |
|      |         | 2.547       | 1.055    |
|      |         | 2.560       | 1.055    |

signals from regular BSS algorithm; (XM BSS out1, out2) present separated signals from XM BSS. The performance evaluation by PESQ is consistent with that measured by SIR. The separation performance evaluated by PESQ and SIR is also consistent with our informal listening tests.

Based on the above simulation, we can see that XM BSS algorithm significantly improves the convergence rate compared with regular time-domain convolutive BSS algorithm.

9. CONCLUSION

In this paper, we investigate time-domain convolutive BSS algorithm and propose two novel algorithms to address the slow convergence rate and high computational complexity problem in time-domain BSS. In the proposed MMax partial update time-domain convolutive BSS algorithm (MMax BSS), only a subset of coefficients in the separation system gets updated at every iteration. We show that the partial update scheme applied in the MMax LMS algorithm for single channel can be extended to multichannel natural gradient-based time-domain convolutive BSS with little deterioration in performance and possible computation complexity saving. In the proposed exclusive maximum selective-tap time-domain convolutive BSS algorithm (XM BSS), the exclusive tap-selection update procedure reduces the interchannel coherence of the tap-input vectors and improves the conditioning of the autocorrelation matrix so as to accelerate convergence rate and reduce the misalignment. Moreover, the computational complexity is reduced as well since only half of tap inputs are selected for updating. Simulation results have shown a significant improvement in convergence rate compared with existing techniques. The extension of the proposed XM BSS algorithm to more than two channels is still an open problem.

REFERENCES

[1] S. Haykin, Ed., Unsupervised Adaptive Filtering, Volume 1: Blind Source Separation, John Wiley & Sons, New York, NY, USA, 2000.
[2] A. Cichocki and S. Amari, Adaptive Blind Signal and Image Processing, John Wiley & Sons, New York, NY, USA, 2000.
[3] A. Hyvärinen, J. Karhunen, and E. Oja, Independent Component Analysis, John Wiley & Sons, New York, NY, USA, 2001.
[4] S. Amari, S. C. Douglas, A. Cichocki, and H. H. Yang, “Multichannel blind deconvolution and equalization using the natural gradient,” in Proceedings of the 1st IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications (SPAWC ’97), pp. 101–104, Paris, France, April 1997.
[5] S. C. Douglas and X. Sun, “Convolutive blind separation of speech mixtures using the natural gradient,” Speech Communication, vol. 39, no. 1-2, pp. 65–78, 2003.
[6] P. Smaragdis, “Blind separation of convolved mixtures in the frequency domain,” Neurocomputing, vol. 22, no. 1–3, pp. 21–34, 1998.
[7] L. Parra and C. Spence, “Convolutive blind separation of non-stationary sources,” IEEE Transactions on Speech and Audio Processing, vol. 8, no. 3, pp. 320–327, 2000.
[8] H. Sawada, R. Mukai, S. Araki, and S. Makino, “A robust and precise method for solving the permutation problem of frequency-domain blind source separation,” IEEE Transactions on Speech and Audio Processing, vol. 12, no. 5, pp. 530–538, 2004.
[9] M. Z. Ikram and D. R. Morgan, “A beamforming approach to permutation alignment for multichannel frequency-domain blind speech separation,” in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’02), vol. 1, pp. 881–884, Orlando, Fla, USA, May 2002.
[10] T. Aboulnasr and K. Mayyas, “Complexity reduction of the NLMS algorithm via selective coefficient update,” IEEE Transactions on Signal Processing, vol. 47, no. 5, pp. 1421–1424, 1999.
[11] A. W. H. Khong and P. A. Naylor, “Stereophonic acoustic echo cancellation employing selective-tap adaptive algorithms,” IEEE Transactions on Audio, Speech and Language Processing, vol. 14, no. 3, pp. 785–796, 2006.
[12] S. Werner, M. L. R. de Campos, and P. S. R. Diniz, “Partial-update NLMS algorithms with data-selective updating,” IEEE Transactions on Signal Processing, vol. 52, no. 4, pp. 938–949, 2004.
[13] I. Pitas, “Fast algorithms for running ordering and max/min calculation,” IEEE Transactions on Circuits and Systems, vol. 36, no. 6, pp. 795–804, 1989.
[14] S. Makino, H. Sawada, R. Mukai, and S. Araki, “Blind source separation of convolutive mixtures of speech in frequency domain,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E88-A, no. 7, pp. 1640–1654, 2005.
[15] ITU-T Recommend P.862, “Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone network and speech codecs,” May 2000.
Research Article

Underdetermined Blind Audio Source Separation Using Modal Decomposition

Abdeldjalil Aïssa-El-Bey, Karim Abed-Meraim, and Yves Grenier

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

Received 1 July 2006; Revised 20 November 2006; Accepted 14 December 2006

This paper introduces new algorithms for the blind separation of audio sources using modal decomposition. Indeed, audio signals and, in particular, musical signals can be well approximated by a sum of damped sinusoidal (modal) components. Based on this representation, we propose a two-step approach consisting of a signal analysis (extraction of the modal components) followed by a signal synthesis (grouping of the components belonging to the same source) using vector clustering. For the signal analysis, two existing algorithms are considered and compared: namely the EMD (empirical mode decomposition) algorithm and a parametric estimation algorithm using ESPRIT technique. A major advantage of the proposed method resides in its validity for both instantaneous and convolutive mixtures and its ability to separate more sources than sensors. Simulation results are given to compare and assess the performance of the proposed algorithms.

Copyright © 2007 Abdeldjalil Aïssa-El-Bey et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

The problem of blind source separation (BSS) consists of finding “independent” source signals from their observed mixtures without a priori knowledge on the actual mixing channels.

The source separation problem is of interest in various applications [1, 2] such as the localization and tracking of targets using radars and sonars, separation of speakers (problem known as “cocktail party”), detection and separation in multiple-access communication systems, independent component analysis of biomedical signals (EEG or ECG), multispectral astronomical imaging, geophysical data processing, and so forth [2].

This problem has been intensively studied in the literature and many effective solutions have been proposed so far [1–3]. Nevertheless, the literature intended for the underdetermined case where the number of sources is larger than the number of sensors (observations) is relatively limited, and achieving the BSS in that context is one of the challenging problems in this field. Existing methods for the underdetermined BSS (UBSS) include the matching pursuit methods in [4, 5], the separation methods for finite alphabet sources in [6, 7], the probabilistic-based (using maximum a posteriori criterion) methods in [8–10], and the sparsity-based techniques in [11, 12]. In the case of nonstationary signals (including the audio signals), certain solutions using time-frequency analysis of the observations exist for the underdetermined case [13–15]. In this paper, we propose an alternative approach named MD-UBSS (for modal decomposition UBSS) using modal decomposition of the received signals [16, 17]. More precisely, we propose to decompose a supposed locally periodic signal which is not necessarily harmonic in the Fourier sense into its various modes. The audio signals, and more particularly the musical signals, can be modeled by a sum of damped sinusoids [18, 19], and hence are well suited for our separation approach. We propose here to exploit this last property for the separation of audio sources by means of modal decomposition. Although we consider here an audio application, the proposed method can be used for any other application where the source signals can be represented by a sum of sinusoidal components. This includes in particular the separation of NMR (nuclear magnetic resonance) signals in [20, 21] and the rotating machine signals in [22]. To start, we consider first the case of instantaneous mixtures, then we treat the more challenging problem of convolutive mixtures in the underdetermined case.
Note that this modal representation of the sources is a particular case of signal sparsity often used to separate the sources in the underdetermined case [23]. Indeed, a signal given by a sum of sinusoids (or damped sinusoids) occupies only a small region in the time-frequency (TF) domain, that is, its TF representation is sparse. This is illustrated by Figure 1 where we represent the time-frequency distribution of a three-modal-component signal.

The paper is organized as follows. Section 2 formulates the UBSS problem and introduces the assumptions necessary for the separation of audio sources using modal decomposition. Section 3 proposes two MD-UBSS algorithms for instantaneous mixture case while Section 4 introduces a modified version of MD-UBSS that relaxes the quasiorthogonality assumption of the source modal components. In Section 5, we extend our MD-UBSS algorithm to the convolutive mixture case. Some discussions on the proposed methods are given in Section 6. The performance of the above methods is numerically evaluated in Section 7. The last section is for the conclusion and final remarks.

2. PROBLEM FORMULATION IN THE INSTANTANEOUS MIXTURE CASE

The blind source separation model assumes the existence of \( N \) independent signals \( s_1(t), \ldots, s_N(t) \) and \( M \) observations \( x_1(t), \ldots, x_M(t) \) that represent the mixtures. These mixtures are supposed to be linear and instantaneous, that is,

\[
x_i(t) = \sum_{j=1}^{N} a_{ij} s_j(t), \quad i = 1, \ldots, M.
\]

This can be represented compactly by the mixing equation

\[
x(t) = As(t),
\]

where \( s(t) \) is an \( N \times 1 \) column vector collecting the real-valued source signals, vector \( x(t) \), similarly, collects the \( M \) observed signals, and the \( M \times N \) mixing matrix \( A \) with \( a_{ij} = [a_{ij}, \ldots, a_{iM}]^T \) contains the mixture coefficients.

Now, if \( N > M \), that is, there are more sources than sensors, we are in the underdetermined case, and BSS becomes UBSS (U stands for underdetermined). By underdeterminacy, we cannot, from the set of equations in (2), algebraically obtain a unique solution, because this system contains more variables (sources) than equations (sensors). In this case, \( A \) is no longer left invertible, because it has more columns than rows. Consequently, due to the underdetermined representation, the above system of (2) cannot be solved completely even with the full knowledge of \( A \), unless we have some specific knowledge about the underlying sources.

Next, we will make some assumptions about the data model in (2), necessary for our method to achieve the UBSS.

**Assumption 1.** The column vectors of \( A \) are pairwise linearly independent.

That is, for any index pair \( i \neq j \in \mathcal{N} \), where \( \mathcal{N} = \{1, \ldots, N\} \), vectors \( a_i \) and \( a_j \) are linearly independent. This assumption is necessary because if otherwise, we have \( a_2 = a_1 a_1 \), for example, then the input/output relation (2) can be reduced to

\[
x(t) = [a_1, a_3, \ldots, a_N][s_1(t) + a_2s_2(t), s_3(t), \ldots, s_N(t))^T,
\]

and hence the separation of \( s_1(t) \) and \( s_2(t) \) is inherently impossible. This assumption is used later (in the clustering step) to separate the source modal components using their spatial directions given by the column vectors of \( A \).

It is known that BSS is only possible up to some scaling and permutation [3]. We take the advantage of these indeterminacies to further make the following assumption without loss of generality.

**Assumption 2.** The column vectors of \( A \) are of unit norm.

That is, \( \|a_i\| = 1 \) for all \( i \in \mathcal{N} \), where the norm hereafter is given in the Frobenius sense.

As mentioned previously, solving the UBSS problem requires strong a priori assumptions on the source signals. In our case, signal sparsity is considered in terms of modal representation of the input signals as stated by the fundamental assumption below.

**Assumption 3.** The source signals are sum of modal components.

Indeed, we assume here that each source signal \( s_i(t) \) is a sum of \( l_i \) modal components \( c_j^i(t) \), \( j = 1, \ldots, l_i \), that is,

\[
s_i(t) = \sum_{j=1}^{l_i} c_j^i(t), \quad t = 0, \ldots, T - 1,
\]

where \( c_j^i(t) \) are damped sinusoids or (quasi)harmonic signals, and \( T \) is the sample size.
Standard BSS techniques are based on the source independence assumption. In the UBSS case, the source independence is often replaced by the disjointness of the sources. This means that there exists a transform domain where the source representation has disjoint or quasidisjoint supports. The quasidisjointness assumption of the sources translates in our case into the quasiorthogonality of the modal components.

Assumption 4. The sources are quasiorthogonal, in the sense that

$$\langle c_i^j \mid c_i^{j'} \rangle \approx 0, \quad \text{for } (i, j) \neq (i', j'),$$

where

$$\langle c_i^j \mid c_i^{j'} \rangle \overset{def}{=} \sum_{t=0}^{T-1} c_i^j(t)c_i^{j'}(t),$$

$$\| c_i^j \|^2 = \langle c_i^j \mid c_i^j \rangle.$$  

In the case of sinusoidal signals, the quasiorthogonality of the modal components is nothing else than the Fourier quasiorthogonality of two sinusoidal components with distinct frequencies. This can be observed in the frequency domain through the disjointness of their supports. This property is also preserved by filtering, which does not affect the frequency support, and hence the quasiorthogonality assumption of the signals (this is used later when considering the convolutive case).

3. MD-UBSS ALGORITHM

Based on the previous model, we propose an approach in two steps consisting of the following.

(i) An analysis step

In this step, one applies an algorithm of modal decomposition to each sensor output in order to extract all the harmonic components from them. We compare for this modal components extraction two decomposition algorithms that are the EMD (empirical mode decomposition) algorithm introduced in [16, 17] and a parametric algorithm which estimates the parameters of the modal components modeled as damped sinusoids.

(ii) A synthesis step

In this step, we group together the modal components corresponding to the same source in order to reconstitute the original signal. This is done by observing that all modal components of a given source signal “live” in the same spatial direction. Therefore, the proposed clustering method is based on the component’s spatial direction evaluated by correlation of the extracted (component) signal with the observed antenna signal.

Algorithm 1: MD-UBSS algorithm in instantaneous mixture case using modal decomposition.

Note that, by this method, each sensor output leads to an estimate of the source signals. Therefore, we end up with $M$ estimates for each source signal. As the quality of source signal extraction depends strongly on the mixture coefficients, we propose a blind source selection procedure to choose the “best” of the $M$ estimates. This algorithm is summarized in Algorithm 1.

3.1. Modal component estimation

3.1.1. Signal analysis using EMD

A new nonlinear technique, referred to as empirical mode decomposition (EMD), has recently been introduced by Huang et al. for representing nonstationary signals as sum of zero-mean AM-FM components [16]. The starting point of the EMD is to consider oscillations in signals at a very local level. Given a signal $z(t)$, the EMD algorithm can be summarized as follows [17]:

1. Extraction of all harmonic components from each sensor by applying modal decomposition.
2. Spatial direction estimation by (14) and vector clustering by $k$-means algorithm [24].
3. Source estimation by grouping together the modal components corresponding to the same spatial direction.
4. Source grouping and source selection by (18).

By applying the EMD algorithm to the $i$th mixture signal $x_i$ which is written as $x_i(t) = \sum_{j=1}^{N_i} a_i^j s_j(t) = \sum_{j=1}^{N_i} a_i^j k_j(t)$, one obtains estimates $\hat{c}_i^j(t)$ of components $c_i^j(t)$ (up to the scalar constant $a_i^j$).

3.1.2. Parametric signal analysis

In this section, we present an alternative solution for signal analysis. For that, we represent the source signal as sum of

\[ z(t) = \sum_{i=1}^{M} \sum_{j=1}^{N_i} a_i^j s_j(t) \]

Indeed, the mean signal $m(t)$ is also the residual signal after extracting the detail component $d(t)$, that is, $m(t) = z(t) - d(t)$. 

\[ m(t) = \sum_{i=1}^{M} \sum_{j=1}^{N_i} a_i^j \]
damped sinusoids:

\[ s_i(t) = \Re\{e^{j\sum_{j=1}^{L} a_j^i (z_j^i)^T}\} ,\]

corresponding to

\[ c_i^j(t) = \Re\{a_j^i (z_j^i)^t\} ,\]

where \( a_j^i = \beta_j^i e^{i\theta_j^i} \) represents the complex amplitude and \( z_j^i = e^{i\omega_j^i t + j\theta_j^i} \) is the \( j \)th pole of the source \( s_i \), where \( d_j^i \) is the negative damping factor and \( \omega_j^i \) is the angular frequency. \( \Re\{\cdot\} \) represents the real part of a complex entity. We denote by \( L_{\text{tot}} \) the total number of modal components, that is, \( L_{\text{tot}} = \sum_{i=1}^{N} l_i \).

For the extraction of the modal components, we propose to use the ESPRIT (estimation of signal parameters via rotational invariance technique) algorithm that estimates the poles of the signals by exploiting the row-shifting invariance property of the \( D \times (T - D) \) data Hankel matrix \( \mathcal{H}(x_k) \) of means by the SVD of \( \mathcal{H}(x_k) \) (\( u_1, \ldots, u_{L_{\text{tot}}} \) are the principal left singular eigenvectors of \( \mathcal{H}(x_k) \));

(3) solve (in the least-squares sense) the shift invariance equation

\[ U_{\text{tot}}^{(L_{\text{tot}})} \Psi = U_{\text{tot}}^{(L_{\text{tot}})} \Psi = U_{\text{tot}}^{(L_{\text{tot}})} \Psi = U_{\text{tot}}^{(L_{\text{tot}})} \Psi ,\]

where \( \Psi = \Phi \Delta \Phi^{-1} \), \( \Phi \) being a nonsingular \( 2L_{\text{tot}} \times 2L_{\text{tot}} \) matrix, and \( \Delta = \text{diag}(z_1^1, z_1^{*1}, \ldots, z_L^1, z_L^{*1}, \ldots, z_N^1, z_N^{*1}) \). \( \cdot^{*} \) represents the complex conjugation, \( \cdot^{T} \) denotes the pseudoinversion operation, and arrows \( \uparrow \) and \( \downarrow \) denote, respectively, the last and the first row-deleting operator;

(4) estimate the poles as the eigenvalues of matrix \( \Psi \);

(5) estimate the complex amplitudes by solving the least-squares fitting criterion

\[ \min_{\alpha_k} \| Z \alpha_k - Z x_k \|^2 \Leftrightarrow \alpha_k = Z^T x_k ,\]

where \( x_k = [x_k(0), \ldots, x_k(T-1)]^T \) is the observation vector, \( Z \) is a Vandermonde matrix constructed from the estimated poles, that is,

\[ Z = [z_1^1, z_1^{*1}, \ldots, z_L^1, z_L^{*1}, \ldots, z_N^1, z_N^{*1}] ,\]

with \( z_j^i = [1, z_j^i, (z_j^i)^2, \ldots, (z_j^i)^{T-1}]^T \), and \( \alpha_k \) is the vector of complex amplitudes, that is,

\[ \alpha_k = \frac{1}{2} [a_k^1 a_1^i, a_k^1 a_1^{*i}, \ldots, a_k^N a_N^{*i}] .\]

**3.2. Clustering and source estimation**

**3.2.1. Signal synthesis using vector clustering**

For the synthesis of the source signals, one observes that thanks to the quasiorthogonality assumption, one has

\[ \langle x_i \mid c_j^i \rangle = \frac{1}{||c_j^i||^2} \sum_j \langle x_i \mid x_j \rangle \approx a_i ,\]

where \( a_i \) represents the \( i \)th column vector of \( A \). We can, then, associate each component \( c_j^i \) to a spatial direction (vector column of \( A \)) that is estimated by

\[ \hat{a}_j^i = \frac{\langle x_i \mid c_j^i \rangle}{||c_j^i||^2} .\]

Vector \( \hat{a}_j^i \) would be equal approximately to \( a_i \) (up to a scalar constant) if \( c_j^i \) is an estimate of a modal component of source \( i \). Hence, two components of a same source signal are associated to colinear spatial direction of to the same column vector of \( A \). Therefore, we propose to gather these components by clustering their directional vectors into \( N \) classes (see Figure 2). For that, we compute first the normalized vectors

\[ \hat{a}_j^i = \frac{\hat{a}_j^i e^{-j\psi_j^i}}{||\hat{a}_j^i||} ,\]

where \( \psi_j^i \) is the phase argument of the first entry of \( \hat{a}_j^i \) (this is to force the first entry to be real positive). Then, these vectors are clustered by \( k \)-means algorithm [24] that can be summarized in the following steps,

(1) Place \( N \) points into the space represented by the vectors that are being clustered. These points represent initial group centroids. One popular way to start is to randomly choose \( N \) vectors among the set of vectors to be clustered.

(2) Assign each vector \( \hat{a}_j^i \) to the group (cluster) that has the closest centroid, that is, if \( y_1, \ldots, y_N \) are the centroids

\[ d(\hat{a}_j^i, y_1), \ldots, d(\hat{a}_j^i, y_N) \]

Figure 2: Data clustering illustration, where we represent the different estimates \( \hat{a}_j^i \) and their centroids.
of the $N$ clusters, one assigns the vector $\mathbf{a}_i^k$ to the cluster $i_0$ that satisfies

$$i_0 = \arg \min_i \| \mathbf{a}_i^k - y_j \|. \quad (16)$$

(3) When all vectors have been assigned, recalculate the positions of the $N$ centroids in the following way: for each cluster, the new centroid’s vector is calculated as the mean value of the cluster’s vectors.

(4) Repeat steps 2 and 3 until the centroids no longer change. This produces a separation of the vectors into $N$ groups. In practice, in order to increase the convergence rate, one can also use a threshold value and stop the algorithm when the difference between the new and old centroid values is smaller than this threshold for all $N$ clusters.

Finally, one will be able to rebuild the initial sources up to a constant by adding the various components within a same class, that is,

$$\hat{s}_i(t) = \sum_{C_i} \hat{\mathbf{c}}_j(t), \quad (17)$$

where $C_i$ is the $i$th cluster.

### 3.2.2. Source grouping and selection

Let us notice that by applying the approach described previously (analysis plus synthesis) to all antenna outputs $x_1(t), \ldots, x_M(t)$, we obtain $M$ estimates of each source signal. The estimation quality of a given source signal varies significantly from one sensor to another. Indeed, it depends strongly on the matrix coefficients and, in particular, on the signal-to-interference ratio (SIR) of the desired source. Consequently, we propose a blind selection method to choose a “good” estimate among the $M$ we have for each source signal. For that, we need first to pair the source estimates together. This is done by associating each source signal extracted from the first sensor to the $(M-1)$ signals extracted from the $(M-1)$ other sensors that are maximally correlated with it. The correlation factor of two signals $s_1$ and $s_2$ is evaluated by $|\langle s_1 | s_2 \rangle|/\|s_1\|\|s_2\|$.

Once the source grouping is achieved, we propose to select the source estimate of maximal energy, that is,

$$\hat{s}_i(t) = \arg \max_{\hat{s}_i(t)} \left\{ E_i^j = \sum_{t=0}^{T-1} |\hat{s}_i^j(t)|^2, \quad j = 1, \ldots, M \right\}, \quad (18)$$

where $E_i^j$ represents the energy of the $i$th source extracted from the $j$th sensor $\hat{s}_i^j(t)$. One can consider other methods of selection (based, e.g., on the dispersion around the centroid) or instead, a diversity combining technique for the different source estimates. However, the source estimates are very dissimilarly in quality, and hence we have observed in our simulations that the energy-based selection, even though not optimal, provides the best results in terms of source estimation error.

### 3.3. Case of common modal components

We consider here the case where a given component $\hat{c}_j^k(t)$ associated with the pole $z_j^k$ can be shared by several sources. This is the case, for example, for certain musical signals such as those treated in [27]. To simplify, we suppose that a component belongs to at most two sources. Thus, let us suppose that the sinusoidal component $(z_j^k)^T$ is present in the sources $s_j(t)$ and $s_j(t)$ with the amplitudes $\alpha_j$ and $\alpha_j$, respectively (i.e., one modal component of source $s_j$ (resp., $s_j$) is $\Re(\alpha_j^z(z_j^k)^T)$ (resp., $\Re(\alpha_j^z(z_j^k)^T)$)). It follows that the spatial direction associated with this component is a linear combination of the column vectors $\mathbf{a}_j$ and $\mathbf{a}_j$. More precisely, we have

$$\hat{\mathbf{a}}_j^k = \frac{1}{\|\mathbf{z}_j^k\|} \begin{bmatrix} x_1^T \mathbf{z}_j^k \\ \vdots \\ x_M^T \mathbf{z}_j^k \end{bmatrix} \approx \alpha_j \mathbf{a}_j + \alpha_j \mathbf{a}_j. \quad (19)$$

It is now a question of finding the indices $j_1$ and $j_2$ of the two sources associated with this component, as well as the amplitudes $\alpha_j$ and $\alpha_j$. With this intention, one proposes an approach based on subspace projection. Let us assume that $M > 2$ and that matrix $\mathbf{A}$ is known and satisfies the condition that any triplet of its column vectors is linearly independent. Consequently, we have

$$\mathbf{P}_\mathbf{A}^\tilde{\mathbf{A}} \hat{\mathbf{a}}_j^k = 0, \quad (20)$$

if and only if $\tilde{\mathbf{A}} = [\mathbf{a}_{j_1}, \mathbf{a}_{j_2}]$, $\tilde{\mathbf{A}}$ being a matrix formed by a pair of column vectors of $\mathbf{A}$ and $\mathbf{P}_\mathbf{A}$ represents the matrix of orthogonal projection on the orthogonal range space of $\tilde{\mathbf{A}}$, that is,

$$\mathbf{P}_\mathbf{A} = \mathbf{I} - \tilde{\mathbf{A}}(\tilde{\mathbf{A}}^H \tilde{\mathbf{A}})^{-1} \tilde{\mathbf{A}}^H, \quad (21)$$

where $\mathbf{I}$ is the identity matrix and $(\cdot)^H$ denotes the transpose conjugate. In practice, by taking into account the noise, one detects the columns $j_1$ and $j_2$ by minimizing

$$\left(j_1, j_2\right) = \arg \min_{(l_m)} \left\{ \|\mathbf{P}_\mathbf{A} \hat{\mathbf{a}}_j^k\| \mid \tilde{\mathbf{A}} = \left[ \mathbf{a}_l \mathbf{a}_m \right] \right\}. \quad (22)$$

Once $\tilde{\mathbf{A}}$ found, one estimates the weightings $\alpha_j$ and $\alpha_j$ by

$$\begin{bmatrix} \alpha_{j_1} \\ \alpha_{j_2} \end{bmatrix} = \tilde{\mathbf{A}}^\dagger \hat{\mathbf{a}}_j^k, \quad (23)$$

In this paper, we treated all the components as being associated to two source signals. If ever a component is present only in one source, one of the two coefficients estimated in (23) should be zero or close to zero.

In what precedes, the mixing matrix $\mathbf{A}$ is supposed to be known. This means that it has to be estimated before applying a subspace projection. This is performed here by clustering all the spatial direction vectors in (14) as for the previous MD-UBSS algorithm. Then, the $i$th column vector of $\mathbf{A}$ is estimated as the centroid of $\mathbf{C}_i$ assuming implicitly that most modal components belong mainly to one source signal. This is confirmed by our simulation experiment shown in Figure 11.
4. MODIFIED MD-UBSS ALGORITHM

We propose here to improve the previous algorithm with respect to the computational cost and the estimation accuracy when Assumption 4 is poorly satisfied. First, in order to avoid repeated estimation of modal components for each sensor output, we use all the observed data to estimate (only once) the poles of the source signals. Hence, we apply the ESPRIT technique on the averaged data covariance matrix $\mathbb{H}(x)$ defined by

$$\mathbb{H}(x) = \sum_{i=1}^{M} \mathcal{H}(x_i) \mathcal{H}(x_i)^H$$

and we apply steps 1 to 4 of Kung’s algorithm described in Section 3.1.2 to obtain all the poles $z^j_i$, $i = 1, \ldots, N$, $j = 1, \ldots, k$. In this way, we reduce significantly the computational cost and avoid the problem of “best source estimate” selection of the previous algorithm.

Now, to relax Assumption 4, we can rewrite the data model as

$$\mathbf{G} \mathbf{z}(t) = \mathbf{x}(t),$$

where $\mathbf{G} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]$, $\mathbf{y}_i = \beta_i e^{j\phi_i} \mathbf{b}_i$ and $\mathbf{y}_i^j = \beta_i e^{-j\phi_i} \mathbf{b}_i$, where $\mathbf{b}_i$ is a unit norm vector representing the spatial direction of the $i$th component (i.e., $\mathbf{b}_i = a_k$ if the component $(z^j_i)^*$ belongs to the $k$th source signal) and $\mathbf{z}(t) = [(z^j_1)^*, (z^j_2)^*, \ldots, (z^j_N)^*, (z^j_N)^*, (z^j_N)^*, (z^j_N)^*]^T$.

The estimation of $\mathbf{G}$ using the least-squares fitting criterion leads to

$$\min_{\mathbf{G}} \| \mathbf{X} - \mathbf{G} \mathbf{Z} \|^2 \iff \mathbf{G} = \mathbf{XZ}^\dagger,$$

where $\mathbf{X} = [\mathbf{x}(0), \ldots, \mathbf{x}(T-1)]$ and $\mathbf{Z} = [\mathbf{z}(0), \ldots, \mathbf{z}(T-1)]$. After estimating $\mathbf{G}$, we estimate the phase of each pole as

$$\phi_i = \arg\left(\frac{\mathbf{y}_i^H \mathbf{y}_i^j}{2}\right).$$

The spatial direction of each modal component is estimated by

$$\hat{\mathbf{a}}_i^j = \mathbf{y}_i^j e^{-j\phi_i} + \mathbf{y}_i^{j*} e^{j\phi_i} = 2\beta_i^j \mathbf{b}_i^j.$$

Finally, we group together these components by clustering the vectors $\hat{\mathbf{a}}_i^j$ into $N$ classes. After clustering, we obtain $N$ classes with $N$ unit-norm centroids $\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_N$ corresponding to the estimates of the column vectors of the mixing matrix $\mathbf{A}$. If the pole $z^j_i$ belongs to the $k$th class, then according to (28), its amplitude can be estimated by

$$\beta_i^j = \frac{\hat{\mathbf{a}}_i^j \hat{\mathbf{a}}_i^j^*}{2}.$$

One will be able to rebuild the initial sources up to a constant by adding the various modal components within a same class $\mathcal{C}_k$ as follows:

$$\hat{s_k}(t) = \Re\left\{ \sum_{\ell \in \mathcal{C}_k} \beta_\ell^j e^{j\phi_\ell^j} (z^j_\ell)^t \right\}.$$

Note that one can also assign each component to two (or more) source signals as in Section 3.3 by using (20)–(23).

5. GENERALIZATION TO THE CONVOLUTIVE CASE

The instantaneous mixture model is, unfortunately, not valid in real-life applications where multipath propagation with large channel delay spread occurs, in which case convolutive mixtures are considered.

Blind separation of convolutive mixtures and multi-channel deconvolution has received wide attention in various fields such as biomedical signal analysis and processing (EEG, MEG, ECG), speech enhancement, geophysical data processing, and data mining [2].

In particular, acoustic applications are considered in situations where signals, from several microphones in a sound field produced by several speakers (the so-called cocktail-party problem) or from several acoustic transducers in an underwater sound field produced by engine noises of several ships (sonar problem), need to be processed.

In this case, the signal can be modeled by the following equation:

$$\mathbf{x}(t) = \sum_{k=0}^{K} \mathbf{H}(k) \mathbf{s}(t - k) + \mathbf{w}(t),$$

where $\mathbf{H}(k)$ are $M \times N$ matrices for $k \in [0, K]$ representing the impulse response coefficients of the channel. We consider in this paper the underdetermined case ($M < N$). The sources are assumed, as in the instantaneous mixture case, to be decomposable in a sum of damped sinusoids satisfying approximately the quasiothogonality Assumption 4. The channel satisfies the following diversity assumption.

Assumption 5. The channel is such that each column vector of

$$\mathbf{H}(z) \overset{\text{def}}{=} \sum_{k=0}^{K} \mathbf{H}(k) z^{-k} \overset{\text{def}}{=} [\mathbf{h}_1(z), \ldots, \mathbf{h}_N(z)]$$

is irreducible, that is, the entries of $\mathbf{h}_i(z)$ denoted by $h_{ij}(z)$, $j = 1, \ldots, M$, have no common zero for all $i$. Moreover, any two column vectors of $\mathbf{H}(z)$ form an irreducible polynomial matrix $\hat{\mathbf{H}}(z)$, that is, rank ($\hat{\mathbf{H}}(z)$) = 2 for all $z$.

Knowing that the convolution preserves the different modes of the signal, we can exploit this property to estimate the different modal components of the source signals using the ESPRIT method considered previously in the instantaneous mixture case. However, using the quasiothogonality assumption, the correlation of a given modal component

\footnote{This is the case when the modal components are closely spaced or for modal components with strong damping factors.}
corresponding to a pole $z^i_k$ of source $s_i$, with the observed signal $x(t)$ leads to an estimate of vector $h_i(z_k)$. Therefore, two components of respective poles $z^i_k$ and $z^j_k$ of the same source signal $s_i$ will produce spatial directions $h_i(z_k)$ and $h_j(z_k)$ that are not colinear. Consequently, the clustering method used for the instantaneous mixture case cannot be applied in this context of convolutive mixtures.

In order to solve this problem, it is necessary to identify first the impulse response of the channels. This problem in overdetermined case is very difficult and becomes almost impossible in the underdetermined case without side information on the considered sources. In this work and similar to [28], we exploit the sparseness property of the audio sources by assuming that from time to time, only one source is present. In other words, we consider the following assumption.

**Assumption 6.** There exist, periodically, time intervals where only one source is present in the mixture. This occurs for all source signals of the considered mixtures (see Figure 3).

To detect these time intervals, we propose to use information criterion tests for the estimation of the number of sources present in the signal (see Section 5.1 for more details). An alternative solution would be to use the “frame selection” technique in [29] that exploits the structure of the spectral density function of the observations. The algorithm in convolutive mixture case is summarized in Algorithm 2.

**5.1. Channel estimation**

Based on Assumption 6, we propose here to apply SIMO- (single-input–multiple-output-) based techniques to blindly estimate the channel impulse response. Regarding the problem at hand, we have to solve 3 different problems: first, we have to select time intervals where only one source signal is effectively present; then, for each selected time interval one should apply an appropriate blind SIMO identification technique to estimate the channel parameters; finally, in the way we proceed, the same channel may be estimated several times and hence one has to group together (cluster) the channel estimates into $N$ classes corresponding to the $N$ source channels.

**5.1.1. Source number estimation**

Let define the spatiotemporal vector

$$x_d(t) = \left[ x^T(t), \ldots, x^T(t-d+1) \right]^T = \sum_{k=1}^{N} H_k s_k(t) + w_d(t),$$

(33)

where $H_k$ are block-Sylvester matrices of size $dM \times (d + K)$ and $s_k(t) \stackrel{\text{def}}{=} [s_k(t), \ldots, s_k(t-K-d+1)]^T$, $d$ is a chosen processing window size. Under the no-common zeros assumption and for large window sizes (see [30] for more details), matrices $H_k$ are full column rank.

Hence, in the noiseless case, the rank of the data covariance matrix

$$R \stackrel{\text{def}}{=} E[x_d(t)x_d^H(t)]$$

is equal to $\min(p(d+K), dM)$, where $p$ is the number of sources present in the considered time interval over which the covariance matrix is estimated. In particular, for $p = 1$, one has the minimum rank value equal to $(d + K)$.

Therefore, our approach consists in estimating the rank of the sample averaged covariance matrix $R$ over several time slots (intervals) and selecting those corresponding to the smallest rank value $r = d + K$.

In the case where $p$ sources are active (present) in the considered time slot, the rank would be $r = p(d+K)$, and hence $p$ can be estimated by the closest integer value to $r/(d+K)$.

**Algorithm 2:** MD-UBSS algorithm in convolutive mixture case using modal decomposition.

1. Channel estimation: AIC criterion [30] to detect the number of sources and application of blind identification algorithm [31, 32] to estimate the channel impulse response.
2. Extraction of all harmonic components from each sensor by applying parametric estimation algorithm (ESPRIT technique).
3. Spatial direction estimation by (44).
4. Source estimation by grouping together, using (45), the modal components corresponding to the same source (channel).
5. Source grouping and source selection by (18).
of $M$ outputs given by

$$x(t) = \sum_{k=0}^{K} h_i(k)s_i(t-k) + w(t),$$

(36)

where $h_i(k) = [h_{i1}(k) \cdots h_{iM}(k)]^T$, $k = 0, \ldots, K$. From (36),
the noise-free outputs $x_j(k)$, $1 \leq j \leq M$, are given by

$$x_j(k) = h_j(k) * s_i(k), \quad 1 \leq j \leq M,$$

(37)

where “$*$” denotes the convolution. Using commutativity of convolution,

it follows that

$$h_{il}(k) * x_j(k) = h_{ij}(k) * x_i(k), \quad 1 \leq j < l \leq M.$$  

(38)

This is a linear equation satisfied by every pair of channels. It was shown
that reciprocally the previous $M(M-1)/2$ cross-relations characterize uniquely
the channel parameters. We have the following theorem [31].

**Theorem 1.** Under the no-common zeros assumption, the set of
cross-relations (in the noise free case):

$$x_i(k) * h_j'(k) - x_j(k) * h_i'(k) = 0, \quad 1 \leq i < j \leq M,$$

(39)

where $h'(z) = [h_1'(z) \cdots h_M'(z)]^T$ is an $M \times 1$ polynomial vector
of degree $K$, is satisfied if and only if $h'(z) = \alpha h_i(z)$ for a given scalar $\alpha$.

By collecting all possible pairs of $M$ channels, one can easily establish a set of linear equations. In matrix form, this
set of equations can be expressed as

$$X_M h_i = 0,$$

(40)

where $h_i \stackrel{\text{def}}{=} [h_{i1}(0) \cdots h_{i1}(K), \ldots, h_{iM}(0) \cdots h_{iM}(K)]^T$ and $X_M$ is defined by

$$X_2 = [X_{(2)} - X_{(1)}],$$

(41)

$$X_n = \begin{bmatrix} X_{(n)} & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & X_{(n)} \end{bmatrix},$$

(41)

with $n = 3, \ldots, M$, and

$$X_{(n)} = \begin{bmatrix} x_n(K) & \cdots & x_n(0) \\ \vdots & \ddots & \vdots \\ x_n(T-K-1) & \cdots & x_n(T-1) \end{bmatrix}.$$  

(42)

In the presence of noise, (40) can be naturally solved in the least-squares (LS) sense according to

$$\hat{h}_i = \arg \min_{h_i} h^H X_M^H X_M h_i,$$

(43)

which solution is given by the least eigenvector of matrix

$X_M^H X_M$. 

---

**Figure 4:** Histogram representing the number of time intervals for each estimated number of sources for 4 audio sources and 3 sensors in convolutive mixture case.

---

**5.1.2. Blind channel identification**

To perform the blind channel identification, we have used in this paper the cross-relation (CR) technique described in
[31, 32]. Consider a time interval where we have only the source $s_i$ present. In this case, we can consider a SIMO system

where $\lambda_i \geq \cdots \geq \lambda_{Md}$ represent the eigenvalues of $R$ and
$T_s$ is the time slot size. Note that it is not necessary at this stage to know exactly the channel degree $K$ as long as $d > K$
(i.e., an overestimation of the channel degree is sufficient) in which case the presence of one source signal is characterized by

$$d < r < 2d.$$  

(35)

Figure 4 illustrates the effectiveness of the proposed method where a recording of 6 seconds of $M = 3$ convolutive mixtures of $N = 4$ sources is considered. The sampling frequency is 8 KHz and the time slot size is $T_s = 200$ samples. The filter coefficients are chosen randomly and the channel order is $K = 6$. One can observe that the case $p = 1$ (one source signal) occurs approximately 10% of the time in the considered context.
Remark 1. We have presented here a basic version of the CR method. In [33], an improved version of the method (introduced in the adaptive scheme) is proposed exploiting the quasisparse nature of acoustic impulse responses.

5.1.3. Clustering of channel vector estimates

The first step of our channel estimation method consists in detecting the time slots where only one single source signal is “effectively” present. However, the same source signal $s_i$ may be present in several time intervals (see Figures 3 and 4) leading to several estimates of the same channel vector $\hat{h}_i$.

We end up, finally, with several estimates of each source channel that we need to group together into $N$ classes. This is done by clustering the estimated vectors using $k$-means algorithm. The $i$th channel estimate is evaluated as the centroid of the $i$th class.

5.2. Component grouping and source estimation

For the synthesis of the source signals, one observes that the quasiorthogonality assumption leads to

$$\hat{h}_i = \frac{\langle x | \hat{c}_i \rangle}{||\hat{c}_i||^2} \propto h_i(z_i^2),$$

where $c_i^2 = e^{d_i + i\omega_i}$ is the pole of the component $\hat{c}_i^2$, that is, $\hat{c}_i^2(t) = \Re e(\hat{c}_i^2(z_i^2)^t)$. Therefore, we propose to gather these components by minimizing the criterion $\alpha$:

$$\hat{c}_i \in C_i \iff i = \arg \min_j \left\{ \min_a ||\hat{h}_i - a h_i(z_i)||^2 \right\},$$

$$i = \arg \min_k \left\{ ||\hat{h}_i||^2 - \frac{||h_i(z_i^2)^2||^2}{||h_i(z_i^2)||^2} \right\},$$

where $h_i$ is the $i$th column of $H$ estimated in Section 5.1 and $h_i(z_i^2)$ is computed by

$$h_i(z_i^2) = \sum_{k=0}^{K} h_i(k)(z_i^2)^{-k}.$$  

One will be able to rebuild the initial sources up to a constant by adding the various components within a same class using (17).

Similar to the instantaneous mixture case, one modal component can be assigned to two or more source signals, which relaxes the quasiorthogonality assumption and improves the estimation accuracy at moderate and high SNRs (see Figure 9).

6. DISCUSSION

We provide here some comments to get more insight onto the proposed separation method.

(i) Overdetermined case

In that case, one is able to separate the sources by left inversion of matrix $A$ (or matrix $H$ in the convolutive case). The latter can be estimated from the centroids of the $N$ clusters (i.e., the centroid of the $i$th cluster represents the estimate of the $i$th column of $A$).

(ii) Estimation of the number of sources

This is a difficult and challenging task in the underdetermined case. Few approaches exist based on multidimensional tensor decomposition [34] or based on the clustering with joint estimation of the number of classes [24]. However, these methods are very sensitive to noise, to the source amplitude dynamic, and to the conditioning of matrix $A$. In this paper, we assumed that the number of sources is known (or correctly estimated).

(iii) Number of modal components

In the parametric approach, we have to choose the number of modal components $L_{tot}$ needed to well-approximate the audio signal. Indeed, small values of $L_{tot}$ lead to poor signal representation while large values of $L_{tot}$ increase the computational cost. In fact, $L_{tot}$ depends on the “signal complexity,” and in general musical signals require less components (for a good modeling) than speech signals [35]. In Section 7, we illustrate the effect of the value of $L_{tot}$ on the separation quality.

(iv) Hybrid separation approach

It is most probable that the separation quality can be further improved using signal analysis in conjunction with spatial filtering or interference cancelation as in [28]. Indeed, it has been observed that the separation quality depends strongly on the mixture coefficients. Spatial filtering can be used to improve the SIR for a desired source signal, and consequently its extraction quality. This will be the focus of a future work.

(v) SIMO versus MIMO channel estimation

We have opted here to estimate the channels using SIMO techniques. However, it is also possible to estimate the channels using overdetermined blind MIMO techniques by considering the time slots where the number of sources is smaller than $(M - 1)$ instead of using only those where the number of “effective” sources is one. The advantage of doing so would be the use of a larger number of time slots (see Figure 4). The drawback resides in the fact that blind identification of MIMO systems is more difficult compared to the SIMO case and leads in particular to higher estimation error (see Figure 12 for a comparative performance evaluation).

\[^3\text{We minimize over the scalar } a \text{ because of the inherent indeterminacy of the blind channel identification, that is, } h_i(z) \text{ is estimated up to a scalar constant as shown by Theorem 1.}\]
In the noiseless case (with perfect modelization of the sources as sums of damped sinusoids), the estimation of the modal components using ESPRIT would be perfect. This would lead to perfect (exact) estimation of the mixing matrix column vectors using least-squares filtering, and hence perfect clustering and source restoration.

7. SIMULATION RESULTS

We present here some simulation results to illustrate the performance of our blind separation algorithms. For that, we consider first an instantaneous mixture with a uniform linear array of \( M = 3 \) sensors receiving the signals from \( N = 4 \) audio sources (except for the third experiment where \( N \) varies in the range \([2 \cdots 6]\)). The angle of arrivals (AOAs) of the sources is chosen randomly.\(^4\) In the convolutive mixture case, the filter coefficients are chosen randomly and the channel order is \( K = 6 \). The sample size is set to \( T = 10000 \) samples (the signals are sampled at a rate of 8 KHz). The observed signals are corrupted by an additive white noise of covariance \( \sigma^2 \mathbf{I} \) (\( \sigma^2 \) being the noise power). The separation quality is measured by the normalized mean-squares estimation errors (NMSEs) of the sources evaluated over \( N_r = 100 \) Monte Carlo runs. The plots represent the averaged NMSE over the

\[ \text{NMSE}_i \left( \alpha \right) = \frac{1}{N_r} \sum_{r=1}^{N_r} \min_{\alpha} \left( \frac{\| \alpha \mathbf{s}_{r,i} - \mathbf{s}_i \|}{\| \mathbf{s}_i \|} \right)^2, \]

where \( \mathbf{s}_i \triangleq [s_i(0), \ldots, s_i(T-1)] \), \( \mathbf{s}_{r,i} \) (defined similarly) is the \( r \)th estimate of source \( \mathbf{s}_i \), and \( \alpha \) is a scalar factor that compensates for the scale indeterminacy of the BSS problem.

In Figure 5, we present a simulation example with \( N = 4 \) audio sources. The upper line represents the original source signals, the second line represents the source estimation by pseudoinversion of mixing matrix \( \mathbf{A} \) assumed exactly known, and the bottom one represents estimates of sources by our algorithm using EMD.

\[ \text{NMSE}_i = \frac{1}{N_r} \sum_{r=1}^{N_r} \text{NMSE}_i, \]

\(^{4}\) This is used here just for the simulation to generate the mixture matrix \( \mathbf{A} \). We do not consider a parametric model using sources AOAs in our separation algorithm.
In other words, there exists an optimal choice of $L$ that depends on the signal type.

In Figure 8, we compare the separation performance loss that we have when the number of sources increases from 2 to 6 in the noiseless case. For $N = 2$ and $N = 3$ (overdetermined case), we estimate the sources by left inversion of the estimate of matrix $A$. In the underdetermined case, the EMD and parametric-based algorithms present similar performance. However, the latter method is better in the overdetermined case.

In Figure 9, we compare the performance of our algorithm using ESPRIT with and without subspace projection. One can observe that using the subspace projection leads to a performance gain at moderate and high SNRs. At low SNRs, the performance is slightly degraded due to the noise effect. Indeed, when a given component belongs "effectively"
to only one source signal, (23) would provide a nonzero amplitude coefficient for the second source due to noise effect which explains the observed degradation.

In Figure 10, we compare the separation performance obtained by our MD-UBSS algorithm and the modified MD-UBSS algorithm. We observe a performance gain in favor of the modified MD-UBSS mainly due to the fact that it does not rely on the quasiorthogonality assumption. This plot also highlights the problem of “best source estimate” selection related to the MD-UBSS as we observe a performance loss between the results given by the proposed energy-based selection procedure and the optimal\(^5\) one using the exact source signals.

Figure 11 illustrates the estimation performance of the mixing matrix \(A\) using proposed clustering method. The observed good estimation performance translates the fact that most modal components belong “effectively” to one single source signal.

In Figure 12, we present the performance of channel identification obtained by using SIMO identification algorithm (in this case, we choose only the time intervals where only one source is present using AIC criterion) with SIMO and MIMO identification algorithms (in this case, we choose the time intervals where we are in the overdetermined case; i.e., where \(p = 1\) or \(p = 2\)). It is observed that SIMO-based identification provides better results than those obtained by SIMO and MIMO identification algorithms.

\(^5\) Clearly, the optimal selection procedure is introduced here just for performance comparison and not as an alternative selection method since it relies on the exact source signals that are unavailable in our context.

\[ CNMSE = 10 \log \frac{\|H - \hat{H}\|^2}{\|H\|^2}. \] (49)
Clearly, the separation quality depends strongly on the quality of channel estimation.

In Figure 14, we present the separation performance when using the exact channel response $H$ compared to that obtained with the proposed estimate $\hat{H}$ using SIMO approach. For SNRs larger than 20 dB, the channel estimation is good enough for the proposed method to achieve almost the same performance as if the channel is exactly known. Surprisingly, at SNR = 20 dB, the channel estimate NMSE is approximately equal to $-18$ dB (see Figure 12), an error level corresponding to a nonnegligible degradation shown in Figure 13. This seemingly contradiction comes from the fact that in the experiment of Figure 13, the channel is disturbed “artificially” using spatially white Gaussian noise, while the real channel estimation error is spatially colored (see, e.g., [37] where explicit expression of the asymptotic channel covariance error is given) which seems to be favorable to our separation method.

8. CONCLUSION

This paper introduces a new blind separation method for audio-type sources using modal decomposition. The proposed method can separate more sources than sensors and provides, in that case, a better separation quality than the one obtained by pseudoinversion of the mixture matrix (even if the latter is known exactly) in the instantaneous mixture case. The separation method proceeds in two steps: an analysis step where all modal components are estimated followed by a synthesis step to group (cluster) together the modal components and reconstruct the source signals. For the signal analysis step, two algorithms are used and compared based, respectively, on the EMD and on the ESPRIT techniques. A modified MD-UBSS as well as a subspace projection approach are also proposed to relax the “quasiothogonaliy” assumption and allow the source signals to share common modal components, respectively. This approach leads to a performance improvement of the separation quality. For the convolutive mixture case, we propose to use again modal decomposition based on ESPRIT technique, but the signal synthesis is more complex and requires the prior identification of the channel impulse response, which is done here using the sparsity of the audio sources.

ACKNOWLEDGMENT

Part of this work has been published in conferences [38, 39].

REFERENCES

[1] A. K. Nandi, Ed., Blind Estimation Using Higher-Order Statistics, Kluwer Academic, Boston, Mass, USA, 1999.
[2] A. Cichocki and S. Amari, Adaptive Blind Signal and Image Processing, John Wiley & Sons, Chichester, UK, 2003.
[3] J.-F. Cardoso, “Blind signal separation: statistical principles,” Proceedings of the IEEE, vol. 86, no. 10, pp. 2009–2025, 1998.
[4] P. Sugden and N. Canagarajah, “Underdetermined noisy blind separation using dual matching pursuits,” in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’04), vol. 5, pp. 557–560, Montreal, Que, Canada, May 2004.
[5] P. Sugden and N. Canagarajah, “Underdetermined blind separation using learned basis function sets,” Electronics Letters, vol. 39, no. 1, pp. 158–160, 2003.
[6] P. Comon, “Blind identification and source separation in 2 × 3 under-determined mixtures,” IEEE Transactions on Signal Processing, vol. 52, no. 1, pp. 11–22, 2004.
[7] A. Belouchrani and J. F. Cardoso, “A maximum likelihood source separation for discrete sources,” in Proceedings of the 7th European Signal Processing Conference (EUSIPCO ’94), vol. 2, pp. 768–771, Scotland, UK, September 1994.

[8] J. M. Peterson and S. Kadambi, “A probabilistic approach for blind source separation of underdetermined convolutive mixtures,” in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’03), vol. 6, pp. 581–584, Hong Kong, April 2003.

[9] S. Y. Low, S. Nordholm, and R. Togneri, “Convolutive blind signal separation with post-processing,” IEEE Transactions on Speech and Audio Processing, vol. 12, no. 5, pp. 539–548, 2004.

[10] L. C. Khor, W. L. Woo, and S. S. Dlay, “Non-sparse approach to underdetermined blind signal estimation,” in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’05), vol. 5, pp. 309–312, Philadelphia, PA, USA, March 2005.

[11] P. Georgiev, F. Theis, and A. Cichocki, “Sparse component analysis and blind source separation of underdetermined mixtures,” IEEE Transactions on Neural Networks, vol. 16, no. 4, pp. 992–996, 2005.

[12] I. Takigawa, M. Kudo, and J. Toyama, “Performance analysis of minimum \( \ell_1 \)-norm solutions for underdetermined source separation,” IEEE Transactions on Signal Processing, vol. 52, no. 3, pp. 582–591, 2004.

[13] N. Linh-Trung, A. Belouchrani, K. Abed-Meraim, and B. Boashash, “Separating more sources than sensors using time-frequency distributions,” EURASIP Journal on Applied Signal Processing, vol. 2005, no. 17, pp. 2828–2847, 2005.

[14] Y. Huang, J. Benesty, and J. Chen, “A blind channel identification-based two-stage approach to separation and dereverberation of speech signals in a reverberant environment,” IEEE Transactions on Speech and Audio Processing, vol. 13, no. 5, pp. 882–895, 2005.

[15] B. Alouby and Y. Deville, “Alternative structures and power spectrum criteria for blind segmentation and separation of convolutive speech mixtures,” in Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA ’03), pp. 361–366, Nara, Japan, April 2003.

[16] M. Wax and T. Kailath, “Detection of signals by information theoretic criteria,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 33, no. 2, pp. 387–392, 1985.

[17] J. Aissa-El-Bey, M. Grebici, K. Abed-Meraim, and A. Be- louchrani, “Blind system identification using cross-relation methods: further results and developments,” in Proceedings of the 7th International Symposium on Signal Processing and Its Applications (ISSPA ’03), vol. 1, pp. 649–652, Paris, France, July 2003.

[18] R. Ahmad, A. W. H. Khong, and P. A. Naylor, “Proportionate frequency domain adaptive algorithms for blind channel identification,” in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’06), vol. 5, pp. 29–32, Toulouse, France, May 2006.

[19] M. A. Boashash, “Separating more sources than sensors using time-frequency masking,” IEEE Transactions on Signal Processing, vol. 43, no. 12, pp. 2828–2847, 1995.

[20] J. Nieuwenhuijsen, R. Heusdens, and Ed. Deprettere, “Robust exponential modeling of audio signals,” in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’98), vol. 6, pp. 3581–3584, Seatttle, Wash, USA, May 1998.

[21] H. Park, S. Van Huffel, and L. Elden, “Fast algorithms for exponential data modeling,” in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’94), vol. 4, pp. 25–28, Adelaide, SA, Australia, April 1994.

[22] C. Serviere, Y. Capdevielle, and J.-L. Lacoume, “Separation of sinusoidal sources,” in Proceedings of IEEE Signal Processing Workshop on Higher-Order Statistics, pp. 344–348, Banff, Canada, July 1997.
[38] A. Aïssa-El-Bey, K. Abed-Meraim, and Y. Grenier, “Blind separation of audio sources using modal decomposition,” in Proceedings of the 8th International Symposium on Signal Processing and Its Applications (ISSPA ’05), vol. 2, pp. 451–454, Sydney, Australia, August 2005.

[39] A. Aïssa-El-Bey, K. Abed-Meraim, and Y. Grenier, “Séparation aveugle sous-déterminée de sources audio par la méthode EMD (Empirical Mode Decomposition),” in Actes 20e Colloque GRETSI sur le Traitement du Signal et des Images, vol. 2, pp. 1233–1236, Louvain-La-Neuve, Belgium, September 2005.