ON THE IMPLEMENTATION OF FULLY ADAPTIVE INTERPOLATED FIR FILTERS

Eduardo L. O. Batista *Rui Seara*

Department of Informatics and Statistics Department of Electrical Engineering
LINSE – Circuits and Signal Processing Laboratory
Federal University of Santa Catarina
88040-900 – Florianópolis – SC – Brazil
E-mails: ebatista@inf.ufsc.br, seara@linse.ufsc.br

ABSTRACT

This paper presents a novel strategy for implementing fully adaptive interpolated finite impulse response (FAIFIR) structures using either the least-mean-square (LMS) or the normalized LMS algorithms. The aim of such a strategy is to mitigate numerical stability issues arising from simultaneously adapting the two cascaded filters (sparse filter and interpolator) that compose a FAIFIR structure. In this context, a modification in the structure of the interpolator is proposed with no impact on both the computational complexity and applicability of the FAIFIR structure. As a result, adaptive filters with enhanced numerical properties are obtained. Numerical simulation results are presented attesting the effectiveness of the proposed strategy.

Index Terms—Adaptive algorithm, adaptive filters, interpolation, least mean square algorithms.

1. INTRODUCTION

The processing capacity of modern digital signal processors (DSPs) has grown considerably over the last decades. As a consequence, signal processing algorithms with higher computational complexity, such as some adaptive filtering algorithms, can now be extensively used in practical applications. On the other hand, several of these algorithms have faced severe complexity constraints because of their increasing use in embedded systems. In this context, the development of effective adaptive filtering algorithms with reduced computational burden is of fundamental importance for supporting several embedded signal processing applications.

Adaptive filtering algorithms very often use a standard finite impulse response (FIR) filter as their underlying structure [1]. However, the computational complexity of this kind of filter can be very high in some applications. Thus, reduced-complexity FIR implementations are frequently used in place of standard FIR filters. One among these implementations is the interpolated FIR (IFIR) filter [2], [3], which has been successfully used for active noise control [4], audio processing in hearing aids [5], echo cancellation [6], and other applications.

The reduced complexity of IFIR structures is obtained by using a sparse FIR filter cascaded with an interpolator (both with reduced coefficient number). In the adaptive context, IFIR filters are implemented either by updating only the coefficients of the sparse filter (adaptive IFIR – AIFIR) [3] or by updating the coefficients from both sparse filter and interpolator (fully adaptive IFIR – FAIFIR) [7]-[10]. The AIFIR approach presents smaller computational burden than the FAIFIR, while the latter commonly lead to better steady-state performance than the former [10]. Since the difference of complexity between such approaches is relatively small [10], the use of the FAIFIR approach is often preferred to the AIFIR. However, the adaptation of FAIFIR structures is somewhat awkward because of the simultaneous update of two cascaded filters (sparse filter and interpolator), which may result in convergence problems that do not occur in AIFIR structures.

This paper presents a novel strategy for implementing FAIFIR filters in which the aforementioned convergence issues are circumvented by modifying the structure of the interpolator. Such a strategy is developed considering the IFIR filter with removed boundary effect (BIFIR), which presents the best performance among the standard IFIR implementations with fixed sparseness [7], [10]. Nevertheless, the proposed strategy can be used as a foundation for developing similar strategies aiming at other FAIFIR implementations [8], [9]. Numerical simulation results are presented aiming to verify the performance of the proposed approach.

This paper is organized as follows. Section 2 describes the main characteristics of BIFIR structures as well as their fully adaptive (FABIFIR) implementation. In Section 3, the proposed strategy to carry out FABIFIR filters is discussed. Section 4 presents simulation results aiming to evaluate the performance of the proposed approach. Finally, concluding remarks are presented in Section 5.

2. FULLY ADAPTIVE IFIR FILTERS WITH REMOVED BOUNDARY EFFECT

Fig. 1 illustrates the block diagram of an IFIR filter with $\mathbf{w}_s$ representing the sparse filter and $\mathbf{g}$, the interpolator. The memory size of the sparse filter is N and its coefficient vector is obtained by setting to zero $L-1$ of each L coefficients of the coefficient vector $\mathbf{w} = [w_0 \;\; w_1 \;\; \cdots$

This work was supported in part by the National Council for Scientific and Technological Development (CNPq).

$w_{N-1}]^{\mathrm{T}}$ of a standard FIR filter, with L denoting the sparseness factor [10]. Thus, the $N\times 1$ coefficient vector of the sparse filter is given by

$$\mathbf{w}_{\mathrm{s}} = [w_0 \;\; 0 \;\cdots\; w_L \;\; 0 \;\cdots\; w_{2L} \;\; 0 \;\cdots\; w_{(N_{\mathrm{s}}-1)L} \;\; 0 \;\cdots\; 0]^{\mathrm{T}} \quad (1)$$

where $N_s = \lfloor (N-1)/L \rfloor + 1$ is the number of nonzero coefficients in (1) with $\lfloor \cdot \rfloor$ denoting the truncation operation. Regarding the interpolator, its coefficient vector is given by

$$\mathbf{g} = [g_0 \;\; g_1 \;\; g_2 \;\cdots\; g_{M-1}]^{\mathrm{T}} \quad (2)$$

with $M = 2L-1$.

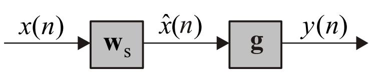

Fig. 1. Block diagram of an IFIR filter.

Also in Fig. 1, $x(n)$ represents the input signal, $\hat{x}(n)$, the input signal filtered by the sparse filter, and

$$y(n) = \mathbf{g}^{\mathrm{T}}\hat{\mathbf{x}}(n) \quad (3)$$

is the output signal with

$$\hat{\mathbf{x}}(n) = [\hat{x}(n) \;\; \hat{x}(n-1) \;\; \hat{x}(n-2) \;\cdots\; \hat{x}(n-M-1)]^{\mathrm{T}} \quad (4)$$

denoting the interpolator input vector. The equivalent coefficient vector of the IFIR filter, which corresponds to its impulse response, is given by

$$\mathbf{w}_{\mathrm{i}} = \mathbf{G}\mathbf{w}_{\mathrm{s}} = \mathbf{W}_{\mathrm{s}}\mathbf{g} \quad (5)$$

where $\mathbf{G}$ and $\mathbf{W}_{\mathrm{s}}$ are convolution matrices [11], with dimensions $N+M-1\times N$ and $N+M-1\times M$, obtained from $\mathbf{g}$ and $\mathbf{w}_{\mathrm{s}}$, respectively. As described in [10], the boundary effect is a characteristic of the equivalent vector $\mathbf{w}_{\mathrm{i}}$ that has a significant impact on the performance of the IFIR filter. To remove such an effect, one pre-multiplies $\mathbf{w}_{\mathrm{i}}$ by a transformation matrix

$$\mathbf{T} = [\,\mathbf{0} \quad \mathbf{I}_N \quad \mathbf{0}] \quad (6)$$

with $\mathbf{0}$ representing a null matrix with dimension $N\times L-1$ and $\mathbf{I}_N$, an identity matrix with dimension $N\times N$ [12]. For illustration purposes, let us consider, for instance, the case of a BIFIR filter with $N=7$ and $L=2$, which implies $M=3$, $\mathbf{g}=[g_0\; g_1\; g_2]^{\mathrm{T}}$, $\mathbf{w}_{\mathrm{s}} = [w_0\; 0\; w_2\; 0\; w_4\; 0\; w_6]^{\mathrm{T}}$, and an equivalent coefficient vector with removed boundary effect given by

$$\begin{aligned}\mathbf{w}_{\mathrm{i}}' = \mathbf{T}\mathbf{w}_{\mathrm{i}} = \mathbf{T}\mathbf{G}\mathbf{w}_{\mathrm{s}} = \{&g_1 w_0 \;\; [g_2 w_0 + g_0 w_2] \;\; g_1 w_2 \\ &[g_2 w_2 + g_0 w_4] \;\; g_1 w_4 \;\; [g_2 w_4 + g_0 w_6] \;\; g_1 w_6\}^{\mathrm{T}}.\end{aligned} \quad (7)$$

Moreover, considering the standard practice of using a triangular window for choosing the interpolator coefficients [3], [10], one obtains $\mathbf{g} = [0.5 \;\; 1 \;\; 0.5]^{\mathrm{T}}$ and

$$\begin{aligned}\mathbf{w}_{\mathrm{i}}' = [w_0 \; &\boxed{0.5w_0 + 0.5w_2} \; w_2 \; \boxed{0.5w_2 + 0.5w_4} \\ &w_4 \; \boxed{0.5w_4 + 0.5w_6} \;\; w_6]^{\mathrm{T}}.\end{aligned} \quad (8)$$

By comparing (1), (7), and (8), we observe that i) the coefficients of the sparse filter $\mathbf{w}_{\mathrm{s}}$ are reproduced in $\mathbf{w}_{\mathrm{i}}'$, and ii) the zero coefficients of $\mathbf{w}_{\mathrm{s}}$ are recreated in $\mathbf{w}_{\mathrm{i}}'$ [see the boxed coefficients in (8)]. The practical implementation of the boundary effect removal is carried out replacing the interpolator input vector $\hat{\mathbf{x}}(n)$ by a modified one given by [10]

$$\hat{\mathbf{x}}'(n) = \mathbf{W}_{\mathrm{s}}^{\mathrm{T}}\mathbf{T}^{\mathrm{T}}\mathbf{x}(n) \quad (9)$$

with $\mathbf{x}(n) = [x(n) \;\; x(n-1) \;\cdots\; x(n-N+1)]^{\mathrm{T}}$, implying a slight increase in the computational burden of $2L-2$ operations per sample, since (9) can be obtained in a recursive way [10].

In the adaptive context, the update expressions to implement FABIFIR filters using the LMS algorithm are [10]

$$\mathbf{g}(n+1) = \mathbf{g}(n) + 2\mu_1 e(n)\underline{\hat{\mathbf{x}}}'(n) \quad (10)$$

and

$$\mathbf{w}_{\mathrm{s}}(n+1) = \mathbf{w}_{\mathrm{s}}(n) + 2\mu_2 e(n)\underline{\tilde{\mathbf{x}}}'(n) \quad (11)$$

where μ_1 and μ_2 denote the step-size parameters and

$$e(n) = d(n) - y(n) \quad (12)$$

is the error signal with $d(n)$ representing the desired signal. Also in (10) and (11), $\underline{\hat{\mathbf{x}}}'(n)$ and $\underline{\tilde{\mathbf{x}}}'(n)$ correspond to approximate versions of $\mathbf{W}_{\mathrm{s}}^{\mathrm{T}}(n)\mathbf{T}^{\mathrm{T}}\mathbf{x}(n)$ and $\mathbf{G}^{\mathrm{T}}(n)\mathbf{T}^{\mathrm{T}}\mathbf{x}(n)$, respectively, both obtained recursively [10].

3. MODIFIED FABIFIR FILTERS

As discussed in [10], among the adaptive realizations of IFIR filters with fixed sparseness characteristics, the FABIFIR implementation is that presenting the best performance in terms of the mean-square error (MSE) at steady state. However, the use of such an implementation is often hampered because of its poor convergence characteristics as compared with adaptive IFIR implementations with fixed interpolators. The main convergence issues of FABIFIR filters arise from simultaneously adapting two cascaded filters. A first drawback is the stalling of the adaptive algorithm if $\mathbf{g}$ and $\mathbf{w}_{\mathrm{s}}$ are zero. In this case, $\mathbf{G}$ and $\mathbf{W}_{\mathrm{s}}$ turn into zero matrices and, consequently, $\underline{\hat{\mathbf{x}}}'(n)$ and $\underline{\tilde{\mathbf{x}}}'(n)$ are zeroed in (10) and (11). A second and more troublesome convergence issue of FABIFIR filters is related to the values attained by the coefficients of vectors $\mathbf{g}$ and $\mathbf{w}_{\mathrm{s}}$ after convergence. Note that the equivalent coefficient vector $\mathbf{w}_{\mathrm{i}}'$ depends on $\mathbf{g}$ and $\mathbf{w}_{\mathrm{s}}$ as described in (7). From such an expression, one notices that the use of coefficient vectors $\bar{\mathbf{g}} = c\mathbf{g}$ and $\bar{\mathbf{w}}_{\mathrm{s}} = (1/c)\mathbf{w}_{\mathrm{s}}$, with c denoting an arbitrary scalar value, results in the same $\mathbf{w}_{\mathrm{i}}'$ as $\mathbf{g}$ and $\mathbf{w}_{\mathrm{s}}$, i.e.,

$$\mathbf{w}'_{\mathrm{i}} = \mathbf{T}\mathbf{G}\mathbf{w}_{\mathrm{s}} = \mathbf{T}\bar{\mathbf{G}}\bar{\mathbf{w}}_{\mathrm{s}} = \mathbf{T}(c\mathbf{G})\left(\mathbf{w}_{\mathrm{s}}/c\right). \tag{13}$$

Then considering (13), we verify that there are infinite combinations of $\mathbf{g}$ and $\mathbf{w}_{\mathrm{s}}$ that produce the same $\mathbf{w}'_{\mathrm{i}}$. Moreover, in situations in which c is much larger than $1/c$, $\mathbf{w}'_{\mathrm{i}}$ can be obtained from a vector $\bar{\mathbf{g}} = c\mathbf{g}$ with large-valued elements and a vector $\bar{\mathbf{w}}_{\mathrm{s}} = (1/c)\mathbf{w}_{\mathrm{s}}$ with small-valued ones. Such characteristic may be particularly undesirable in implementations using fixed-point arithmetic. In these cases, small values may imply loss of precision and large values may exceed the maximum value supported by the adopted numerical representation (overflow). Thus, one has an important threat for both numerical stability and algorithm convergence.

Aiming to improve the convergence characteristics of the FABIFIR filter, a new strategy for updating the coefficients is proposed in this paper. The main idea is to exploit the role of the central coefficient of the interpolator in the FABIFIR structure. Such a coefficient has the specific task of reproducing the coefficients of the sparse filter in the equivalent structure, which can be verified comparing (7) and (8) as well as observing that g_1 is a multiplier of the coefficients of the sparse filter in (7). Thus, in the adaptive process, one can fix the central coefficient of the interpolator to an arbitrary value k without loss of generality in the equivalent structure. Such a strategy prevents the previously mentioned stalling of the adaptive process, since the interpolator coefficient vector $\mathbf{g}$ always presents at least one nonzero coefficient. Additionally, the numerical stability issue is also overcome, since a given $\mathbf{w}'_{\mathrm{i}}$ can only be obtained from a unique combination of the sparse filter coefficient vector $\mathbf{w}_{\mathrm{s}}$ and the interpolator coefficient vector with central coefficient fixed to k (denoted here by $\mathbf{g}_{[k]}$). Such a feature is discussed in the next section.

3.1. Uniqueness of the Proposed Solution

If the proposed strategy of fixing the central coefficient of the interpolator to k leads to unique vectors $\mathbf{w}_{\mathrm{s}}$ and $\mathbf{g}_{[k]}$ for a given $\mathbf{w}'_{\mathrm{i}}$, the equality

$$\mathbf{w}'_{\mathrm{i}} = \mathbf{T}\mathbf{G}_{[k]\mathrm{a}}\mathbf{w}_{\mathrm{sa}} = \mathbf{T}\mathbf{G}_{[k]\mathrm{b}}\mathbf{w}_{\mathrm{sb}} \tag{14}$$

must only be upheld for $\mathbf{w}_{\mathrm{sa}} = \mathbf{w}_{\mathrm{sb}}$ and $\mathbf{g}_{[k]\mathrm{a}} = \mathbf{g}_{[k]\mathrm{b}}$. In (14), $\mathbf{w}_{\mathrm{sa}}$ and $\mathbf{w}_{\mathrm{sb}}$ are generic sparse vectors in the form of (1), while $\mathbf{G}_{[k]\mathrm{a}}$ and $\mathbf{G}_{[k]\mathrm{b}}$ are convolution matrices obtained from generic vectors $\mathbf{g}_{[k]\mathrm{a}}$ and $\mathbf{g}_{[k]\mathrm{b}}$ with the central coefficient set to k. By considering (6) and the structure of $\mathbf{G}_{[k]}$, we notice that $\mathbf{T}\mathbf{G}_{[k]}$ is an $N \times N$ matrix with all elements in the main diagonal equal to k. Thus,

$$\mathbf{T}\mathbf{G}_{[k]} = k\mathbf{I}_N + \underline{\mathbf{G}}_{[k]} \tag{15}$$

where $\mathbf{I}_N$ is an identity matrix with dimension $N \times N$ and $\underline{\mathbf{G}}_{[k]}$, a matrix obtained zeroing the elements in the main diagonal of $\mathbf{T}\mathbf{G}_{[k]}$. Then, considering (15), (14) can be rewritten as

$$\mathbf{w}'_{\mathrm{i}} = k\mathbf{w}_{\mathrm{sa}} + \underline{\mathbf{G}}_{[k]\mathrm{a}}\mathbf{w}_{\mathrm{sa}} = k\mathbf{w}_{\mathrm{sb}} + \underline{\mathbf{G}}_{[k]\mathrm{b}}\mathbf{w}_{\mathrm{sb}}. \tag{16}$$

By analyzing each of the terms from (16), we verify that a) $k\mathbf{w}_{\mathrm{sa}}$ and $k\mathbf{w}_{\mathrm{sb}}$ are vectors with elements different from zero only in the positions whose indices are multiple of k [see (1)], and b) the vectors resulting from $\underline{\mathbf{G}}_{[k]\mathrm{a}}\mathbf{w}_{\mathrm{sa}}$ and $\underline{\mathbf{G}}_{[k]\mathrm{b}}\mathbf{w}_{\mathrm{sb}}$ are also sparse with nonzero coefficients only in the positions whose indices are not multiples of k. Thus, (16) can be split into two terms, i.e.,

$$k\mathbf{w}_{\mathrm{sa}} = k\mathbf{w}_{\mathrm{sb}} \tag{17}$$

and $\underline{\mathbf{G}}_{[k]\mathrm{a}}\mathbf{w}_{\mathrm{sa}} = \underline{\mathbf{G}}_{[k]\mathrm{b}}\mathbf{w}_{\mathrm{sb}}$. Since k is a scalar, (17) and consequently (14) can only be fulfilled if $\mathbf{w}_{\mathrm{sa}} = \mathbf{w}_{\mathrm{sb}}$. Thus, one verifies that there is a unique vector $\mathbf{w}_{\mathrm{s}}$ for a given $\mathbf{w}'_{\mathrm{i}}$. Thereby, considering (5), (7), and also that $\mathbf{w}_{\mathrm{sa}} = \mathbf{w}_{\mathrm{sb}}$, (14) can be rewritten as

$$\mathbf{w}'_{\mathrm{i}} = \mathbf{T}\mathbf{W}_{\mathrm{sa}}\mathbf{g}_{[k]\mathrm{a}} = \mathbf{T}\mathbf{W}_{\mathrm{sb}}\mathbf{g}_{[k]\mathrm{b}} = \mathbf{T}\mathbf{W}_{\mathrm{sa}}\mathbf{g}_{[k]\mathrm{b}}. \tag{18}$$

In practice, $\mathbf{T}\mathbf{W}_{\mathrm{sa}}$ has dimension $N \times M$ and a full rank equal to M. Therefore, $\mathbf{T}\mathbf{W}_{\mathrm{sa}}$ presents a left inverse [12], denoted here by $(\mathbf{T}\mathbf{W}_{\mathrm{sa}})^{+}$, in such a way that $(\mathbf{T}\mathbf{W}_{\mathrm{sa}})^{+}\mathbf{T}\mathbf{W}_{\mathrm{sa}} = \mathbf{I}_M$, with $\mathbf{I}_M$ representing an $M \times M$ identity matrix. Thus, by pre-multiplying the terms of (18) by $(\mathbf{T}\mathbf{W}_{\mathrm{sa}})^{+}$, we get

$$\mathbf{g}_{[k]\mathrm{a}} = \mathbf{g}_{[k]\mathrm{b}}. \tag{19}$$

From (19), one observes that a given $\mathbf{w}'_{\mathrm{i}}$ can only be obtained from a unique $\mathbf{g}_{[k]}$. As a result, we verify the uniqueness of both $\mathbf{w}_{\mathrm{s}}$ and $\mathbf{g}_{[k]}$ for a given $\mathbf{w}'_{\mathrm{i}}$.

3.2. Interpolator Coefficient Update

Aiming to obtain the coefficient update equation for both the interpolator and sparse filter, a constrained approach is considered here [3], [11]. In the case of the interpolator, the constraint resulting from fixing the central coefficient to k is obtained as

$$\mathbf{c}_{\mathrm{i}}^{\mathrm{T}}\mathbf{g}_{[k]}(n) = k \tag{20}$$

where

$$\mathbf{c}_{\mathrm{i}} = [\underbrace{0 \ \cdots \ 0}_{L-1 \text{ elements}} \ 1 \ \underbrace{0 \ \cdots \ 0}_{L-1 \text{ elements}}]^{\mathrm{T}} \tag{21}$$

is the constraint vector. In the case of using the LMS algorithm, the update equation is

$$\mathbf{g}_{[k]}(n+1) = \mathbf{g}_{[k]}(n) - \mu_{\mathrm{g}}\nabla_{\mathbf{g}_{[k]}}J(n) \tag{22}$$

with μ_{g} representing the step-size parameter and $\nabla_{\mathbf{g}_{[k]}}J(n)$, the gradient of a cost function obtained by including the constraint, using the Lagrange multiplier method [11], to the instantaneous estimate of the MSE. The cost function is given by

$$J(n) = e^2(n) + \theta(n)[\mathbf{c}_i^T \mathbf{g}_{[k]}(n) - k] \quad (23)$$

where $\theta(n)$ denotes the Lagrange multiplier. By evaluating $\nabla_{\mathbf{g}_{[k]}} J(n)$, substituting the resulting expression into (22), solving for the Lagrange multiplier similarly to [11], and considering the recursive form of computing $\hat{\underline{\mathbf{x}}}'(n) \cong \mathbf{W}_s^T(n)\mathbf{T}^T\mathbf{x}(n)$, the following expression for adapting the interpolator is obtained:

$$\mathbf{g}_{[k]}(n+1) = \mathbf{P}_i \mathbf{g}_{[k]}(n) + 2\mu_i e(n)\mathbf{P}_i \hat{\underline{\mathbf{x}}}'(n) + k\mathbf{c}_i \quad (24)$$

with $\mathbf{P}_i = \mathbf{I}_M - \mathbf{c}_i \mathbf{c}_i^T$. The computational cost to implement (24) is slightly smaller than the cost to obtain (10) (standard FABIFIR case) mainly because of the pre-multiplication of some right-hand side (R.H.S.) terms of (24) by $\mathbf{P}_i$, which implies to update all interpolator coefficients except the central one. In the case of the NLMS algorithm, the coefficient update equation is obtained by minimizing the Euclidian norm of

$$\delta\mathbf{g}_{[k]}(n+1) = \mathbf{g}_{[k]}(n+1) - \mathbf{g}_{[k]}(n) \quad (25)$$

under the following constraints:

$$\mathbf{g}_{[k]}(n+1)\mathbf{W}_s^T \mathbf{T}^T \mathbf{x}(n) = d(n) \quad (26)$$

and

$$\mathbf{c}_i^T \mathbf{g}_{[k]}(n+1) = k. \quad (27)$$

Then, using the Lagrange multiplier method to include the constraints in the cost function, evaluating the gradient of such a cost function with respect to $\mathbf{g}_{[k]}(n+1)$, solving the resulting expression for the Lagrange multipliers as in [11], and considering the recursive computation of $\hat{\underline{\mathbf{x}}}'(n) \cong \mathbf{W}_s^T(n)\mathbf{T}^T\mathbf{x}(n)$, the following update expression for the NLMS algorithm is obtained:

$$\mathbf{g}_{[k]}(n+1) = \mathbf{P}_i \mathbf{g}_{[k]}(n) + \frac{\alpha_g}{\|\mathbf{P}_i \hat{\underline{\mathbf{x}}}'(n)\|^2 + \psi_g} e(n)\mathbf{P}_i \hat{\underline{\mathbf{x}}}'(n) + k\mathbf{c}_i \quad (28)$$

where α_g denotes the control parameter and $\psi_g > 0$ is a regularization parameter that prevents division by zero.

3.3. Sparse Filter Coefficient Update

The expressions to update the coefficients of the sparse filter are obtained considering that the constraints arising from the sparseness of such a filter can be described as [3], [11]

$$\mathbf{C}^T \mathbf{w}_s(n) = \mathbf{f} \quad (29)$$

where $\mathbf{C}$ is the constraint matrix and $\mathbf{f}$, a response vector. Following the steps presented in [11], one obtains the following expression for updating the sparse filter using the LMS algorithm:

$$\mathbf{w}_s(n+1) = \mathbf{P}\mathbf{w}_s(n) + 2\mu_w e(n)\mathbf{P}\tilde{\underline{\mathbf{x}}}'(n) \quad (30)$$

with $\mathbf{P} = \mathbf{I}_N - \mathbf{C}\mathbf{C}^T$ and μ_w denoting the step size. In addition, for the NLMS algorithm, again following the steps from [11], one obtains

$$\mathbf{w}_s(n+1) = \mathbf{P}\mathbf{w}_s(n) + \frac{\alpha_w}{\|\mathbf{P}\tilde{\underline{\mathbf{x}}}'(n)\|^2 + \psi_w} e(n)\mathbf{P}\tilde{\underline{\mathbf{x}}}'(n) \quad (31)$$

where α_w and ψ_w represent control parameters similar to α_g and ψ_g, respectively.

4. Simulation Results

In this section, simulation results are shown aiming to assess the performance of the modified FABIFIR (MFABIFIR) implementation (proposed approach) as compared with the FABIFIR approach proposed in [10]. Thus, FABIFIR and MFABIFIR filters are applied to a system modeling problem and compared firstly in terms of the MSE. The MSE curves are obtained from Monte Carlo simulations (average of 100 independent runs). The FABIFIR and MFABIFIR structures are also confronted in terms of the quadratic norms of $\mathbf{w}_s$ and $\mathbf{g}$ at the end of each independent run. The goal of such a comparison is to verify the fluctuation of $\mathbf{w}_s$ and $\mathbf{g}$ after convergence. Here, the FABIFIR and MFABIFIR approaches consider a sparseness factor $L = 2$, an initial value of the interpolator coefficient vector given by $\mathbf{g}(0) = [0\ 1\ 0]^T$, and memory size 71. The central coefficient of the interpolator from the MFABIFIR structure is fixed to $k = 1$. The adaptive algorithm is the NLMS with $\alpha_g = \alpha_w = 0.5$ and $\psi_g = \psi_w = 1$. The input signal is white and Gaussian with unity variance, and the noise (added to the output of the plant) is also white and Gaussian with variance $\sigma_z^2 = 10^{-8}$ (80 dB SNR). In addition, the plant presents the impulse response shown in Fig. 2 (memory size 71).

The MSE curves obtained considering the simulation scenario described above are shown in Fig. 3. There, we notice the similar performance of the FABIFIR and MFABIFIR filters in steady state, attesting that both filters get nearly the same equivalent coefficient vector. On the other hand, from the final quadratic norms of $\mathbf{w}_s$ and $\mathbf{g}$ illustrated in Fig. 4, we observe a considerable fluctuation for the FAIFIR filter (dark and gray solid lines), while the fluctuation is unnoticeable for the proposed MFABIFIR filter (dashed and dotted lines).

To assess the performance of the FABIFIR and the proposed MFABIFIR filter under numerical constraints, we consider a second simulation scenario in which we limit the absolute value of each coefficient to 1.5 (saturation constraints). Note that these constraints are very mild considering the values of the plant coefficients (the larger one is equal to one) and also the constraints found in practical applications (where the coefficients and signal samples often present bounded values and limited numerical precision). The MSE curves obtained using the saturation constraints are shown in Fig. 5. We can observe the poor performance of the FABIFIR filter as compared with the proposed MFABIFIR, which presented a result as good as the one obtained without any numerical constraints. From the results of final quadratic norms shown in Fig. 6, one observes smaller but still considerable fluctuations of the norms of $\mathbf{w}_s$ and $\mathbf{g}$ for the FABIFIR filter, while the fluctuations are again unnoticeable for the MFABIFIR.

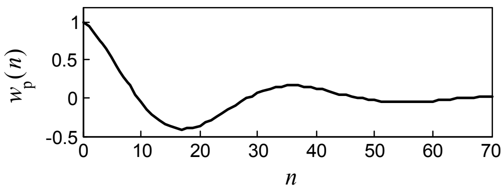

Fig. 2. Impulse response of the plant.

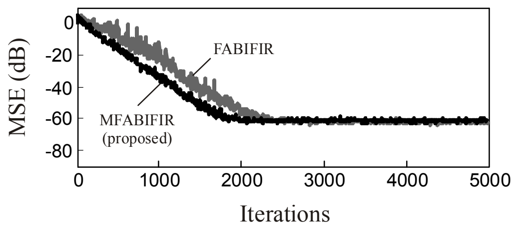

Fig. 3. MSE curves obtained without using numerical constraints.

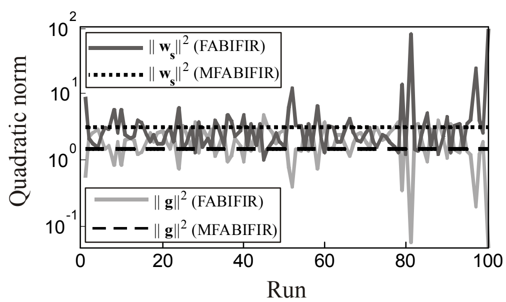

Fig. 4. Quadratic norms obtained at the end of each independent run without the use of numerical constraints.

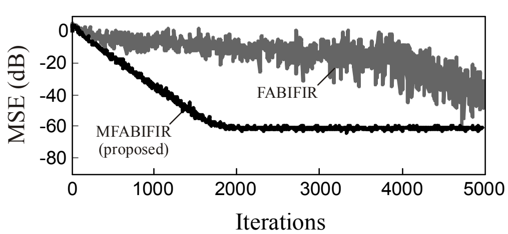

Fig. 5. MSE curves obtained using numerical constraints.

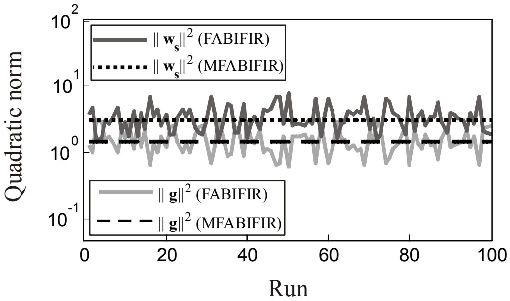

Fig. 6. Quadratic norms obtained at the end of each independent run using numerical constraints.

5. CONCLUDING REMARKS

In this paper, a novel effective strategy for implementing fully adaptive IFIR filters was discussed. Such a strategy is based on modifying the structure of the interpolator aiming to overcome convergence issues and to ensure numerical stability. The obtained implementation outperforms the conventional fully adaptive IFIR in terms of convergence characteristics and numerical properties, as well as exhibits a slightly smaller computational cost. Numerical simulation results were presented attesting the effectiveness of the proposed strategy.

6. REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*, 4 ed., Prentice-Hall, 2002.

[2] Y. Neuvo, C. Y. Dong, and S. K. Mitra, "Interpolated finite impulse response digital filters," *IEEE Trans. Acoust., Speech, Signal Process*, vol. 32, no. 3, pp. 563-570, Jun. 1984.

[3] O. J. Tobias and R. Seara, "Analytical model for the first and second moments of an adaptive interpolated FIR filter using the constrained filtered-X LMS algorithm," *IEE Proc.- Vis., Image, Signal Process.*, vol. 148, no. 5, pp. 337-347, Oct. 2001.

[4] K. Rajgopal and S. Venkataraman, "A delayless adaptive IFIR filterbank structure for wideband and narrowband active noise control," *Signal Processing*, vol. 86, no. 11, pp. 3421-3431, Nov. 2006.

[5] S. Nielsen and J. Sparso, "Designing asynchronous circuits for low power: An IFIR filter bank for a digital hearing aid," *Proceedings of the IEEE*, vol. 87, no. 2, pp. 268-281, Feb. 1999.

[6] S.-S. Lin and W.-R. Wu, "A low-complexity adaptive echo canceller for xDSL Applications," *IEEE Trans. Signal Process.*, vol. 52, no. 5, pp. 1461-1465, May. 2004.

[7] R. C. Bilcu, P. Kuosmanen, and K. Egiazarian, "On adaptive interpolated FIR filters," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Montreal, Canada, vol. 2, May 2004, pp. 665-668.

[8] R. C. de Lamare and R. Sampaio-Neto, "Adaptive reduced-rank MMSE filtering with interpolated FIR filters and adaptive interpolators," *IEEE Signal Process. Letters*, vol. 12, no. 3, pp. 177-180, Mar. 2005.

[9] R. C. de Lamare and R. Sampaio-Neto, "Adaptive reduced-rank processing based on joint an iterative interpolation, decimation and filtering", *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2503-2514, Jul. 2009.

[10] E. L. O. Batista, O. J. Tobias, and R. Seara, "A fully adaptive IFIR filter with removed border effect," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Las Vegas, USA, Apr. 2008, pp. 3821-3824.

[11] E. L. O. Batista, O. J. Tobias, and R. Seara, "A sparse-interpolated scheme for implementing adaptive Volterra filters," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2022-2035, Apr. 2010.

[12] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas with Application to Linear Systems Theory*. Princeton, NJ: Princeton University Press, 2005.