A FULLY ADAPTIVE IFIR FILTER WITH REMOVED BORDER EFFECT

Eduardo L. O. Batista, Orlando J. Tobias, and Rui Seara

LINSE – Circuits and Signal Processing Laboratory
Department of Electrical Engineering
Federal University of Santa Catarina
88040-900 – Florianópolis – SC – Brazil
E-mails: {dudu, orlando, seara}@linse.ufsc.br

ABSTRACT

This paper presents a procedure for implementing fully adaptive interpolated FIR filters with removed border effect. The proposed approach allows reducing the steady-state mean-square error by eliminating the main sources of performance degradation from the adaptive interpolated FIR filters. In addition, the computational effort needed for implementing such a procedure is very small. Simulation results confirm the effectiveness of the proposed approach.

***Index Terms*—**Adaptive filters, adaptive signal processing, interpolation, least mean square methods.

1. INTRODUCTION

Interpolated finite impulse response (IFIR) filters are a computationally efficient alternative to implement FIR filters [1]. The idea of such a scheme is to realize a FIR filter through a cascade structure composed of a sparse filter, with reduced number of coefficients, and an interpolator filter. The latter one recreates (in an approximate way) the removed coefficients of the sparse filter. Different adaptive implementations of IFIR filters have been considered in the open literature for several applications, such as line echo canceling [2]-[4], active noise and vibration control [5], and audio processing in digital hearing aids [6].

The use of adaptive IFIR (AIFIR) filters is of particular interest due to the reduced number of coefficients to be adapted as compared with the standard FIR approach. However, the obtained computational saving gives rise to a higher steady-state mean-square error (MSE) value. This poorer performance can be amended by using adaptive interpolators instead of fixed ones, resulting in a fully adaptive IFIR structure [7]-[9]. Another source of performance degradation in IFIR filters is the border effect, arising from the equivalent IFIR structure [10]. A procedure discussed in [11] permits to remove such an effect in AIFIR filters as using fixed interpolators, resulting in a considerable performance improvement. In this work, a procedure to implement fully adaptive IFIR filters with removed border effect is presented. The focus of such an approach is to improve the AIFIR structure performance.

This paper is organized as follows. Section 2 describes briefly the IFIR filters, their implementations by considering input or output interpolators, and their equivalent structure. In Section 3, a generalized procedure for removing the border effect in IFIR structures is discussed. Section 4 describes the fully adaptive IFIR implementation with removed border effect. Section 5 presents some numerical simulations. Finally, Section 6 draws the conclusions of this research work.

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

2. IFIR FILTERS

The block diagram of an IFIR filter is shown in Fig. 1, where $x(n)$ and $y(n)$ represent, respectively, the input and output signals. The interpolator, denoted by $\mathbf{i}$, consists of a FIR filter with a memory size M and a coefficient vector given by

$$\mathbf{i} = [i_0 \;\; i_1 \;\; \cdots \;\; i_{M-1}]^{\mathrm{T}} . \tag{1}$$

The interpolator output $\tilde{x}(n)$ is related to the input signal by

$$\tilde{x}(n) = x(n) * \mathbf{i} \tag{2}$$

where "$*$" denotes the convolution operation. The block $\mathbf{w}_{\mathrm{s}}$ in Fig. 1 represents a sparse filter with memory size N. The coefficient vector for such a filter is obtained by setting to zero $L-1$ of each L coefficients of a standard FIR filter (L denotes the interpolation factor). Thus, by considering the coefficient vector of a standard FIR filter given by

$$\mathbf{w} = [w_0 \;\; w_1 \;\; w_2 \;\; \cdots \;\; w_{N-1}]^{\mathrm{T}} \tag{3}$$

the following sparse coefficient vector can then be obtained:

$$\mathbf{w}_{\mathrm{s}} = [w_0 \;\; 0 \; \cdots \; w_L \;\; 0 \; \cdots \; w_{2L} \;\; 0 \; \cdots \; w_{(N_{\mathrm{s}}-1)L} \;\; 0 \; \cdots \; 0]^{\mathrm{T}}. \tag{4}$$

The number of nonzero coefficients in (4) is

$$N_{\mathrm{s}} = \lfloor (N-1)/L \rfloor + 1 \tag{5}$$

with $\lfloor \cdot \rfloor$ representing the truncation operation. The output signal of the structure is obtained as follows:

$$y(n) = \tilde{x}(n) * \mathbf{w}_{\mathrm{s}} = \tilde{\mathbf{x}}^{\mathrm{T}}(n)\mathbf{w}_{\mathrm{s}} \tag{6}$$

with the interpolated input vector given by

$$\tilde{\mathbf{x}}(n) = [\tilde{x}(n) \;\; \tilde{x}(n-1) \;\; \tilde{x}(n-2) \;\; \cdots \;\; \tilde{x}(n-N+1)]^{\mathrm{T}}. \tag{7}$$

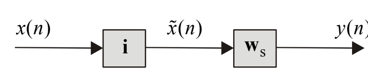

Fig. 1. Block diagram of an IFIR filter.

Another alternative IFIR implementation is shown in Fig. 2. Note that in this one the blocks of Fig. 1 are interchanged. In Fig. 2, $\hat{x}(n)$ represents the input signal filtered by the sparse filter. Now, the output signal is

$$y(n) = \hat{x}(n) * \mathbf{i} = \hat{\mathbf{x}}^{\mathrm{T}}(n)\,\mathbf{i} \tag{8}$$

with

$$\hat{\mathbf{x}}(n) = [\hat{x}(n)\ \ \hat{x}(n-1)\ \ \hat{x}(n-2)\ \ \cdots\ \ \hat{x}(n-M+1)]^{\mathrm{T}} \tag{9}$$

and

$$\hat{x}(n) = x(n) * \mathbf{w}_{\mathrm{s}}. \tag{10}$$

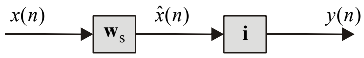

Fig. 2. Block diagram of an IFIR filter with output interpolator.

Both IFIR implementations (Figs. 1 and 2) share the same $(N+M-1)$-dimensional equivalent coefficient vector, given by

$$\mathbf{w}_{\mathrm{i}} = \mathbf{i} * \mathbf{w}_{\mathrm{s}}\ . \tag{11}$$

As described in [10], it is possible to rewrite (11) as a matrix-vector product; then, by defining an $[(N+M-1)\times N]$-dimensional interpolation matrix as

$$\mathbf{I} = \begin{bmatrix} i_0 & 0 & 0 & \cdots & 0 \\ i_1 & i_0 & 0 & \cdots & 0 \\ i_2 & i_1 & i_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ i_{M-1} & i_{M-2} & i_{M-3} & \cdots & i_0 \\ 0 & i_{M-1} & i_{M-2} & \cdots & i_1 \\ 0 & 0 & i_{M-1} & \cdots & i_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & i_{M-1} \end{bmatrix} \tag{12}$$

from (11) and (12), one can write

$$\mathbf{w}_{\mathrm{i}} = \mathbf{I}\mathbf{w}_{\mathrm{s}}. \tag{13}$$

A matrix $\mathbf{W}_{\mathrm{s}}$, with dimensions $(N+M-1)\times M$, analogous to (12), can be constructed from (4) [12], such that

$$\mathbf{w}_{\mathrm{i}} = \mathbf{I}\mathbf{w}_{\mathrm{s}} = \mathbf{W}_{\mathrm{s}}\mathbf{i}\ . \tag{14}$$

Further, by considering an input vector with $N+M-1$ samples of the input signal, given by

$$\mathbf{x}_{\mathrm{e}}(n) = \left[x(n)\ \ x(n-1)\ \ \cdots\ \ x(n-N-M+2)\right]^{\mathrm{T}} \tag{15}$$

the overall input-output relationship for the IFIR filter can be expressed as

$$y(n) = \mathbf{x}_{\mathrm{e}}^{\mathrm{T}}(n)\mathbf{I}\mathbf{w}_{\mathrm{s}} = \mathbf{x}_{\mathrm{e}}^{\mathrm{T}}(n)\mathbf{W}_{\mathrm{s}}\mathbf{i}\ . \tag{16}$$

Note that (14), (15), and (16) are valid for both implementations considered in Fig. 1 and 2.

The task of the interpolator filter in an IFIR structure is to recreate the zeroed coefficients in (4) by using their neighbors [10]. The coefficient number M of the interpolator is a function of factor L, obtained from

$$M(L) = 1 + 2(L-1) = 2L - 1\ . \tag{17}$$

For instance, an interpolation factor $L=2$ results in $M=3$ and $\mathbf{i} = \left[i_0\ \ i_1\ \ i_2\right]^{\mathrm{T}}$. Thus, for a sparse filter considering $N=5$, the coefficient vector is given by $\mathbf{w}_{\mathrm{s}} = [w_0\ \ 0\ \ w_2\ \ 0\ \ w_4]^{\mathrm{T}}$ and the equivalent coefficient vector, obtained from (14), is

$$\mathbf{w}_{\mathrm{i}} = \left[i_0w_0\ \ i_1w_0\ \ (i_2w_0 + i_0w_2)\ \ i_1w_2\ \ (i_2w_2 + i_0w_4)\ \ i_1w_4\ \ i_2w_4\right]^{\mathrm{T}}. \tag{18}$$

Then, by using a linear interpolator $\mathbf{i} = [0.5\ \ 1\ \ 0.5]^{\mathrm{T}}$ in (18), results in [1]

$$\mathbf{w}_{\mathrm{i}} = \left[\underline{0.5w_0}\ \ w_0\ \ \boxed{0.5w_0 + 0.5w_2}\ \ w_2\ \ \boxed{0.5w_2 + 0.5w_4}\ \ w_4\ \ \underline{0.5w_4}\right]^{\mathrm{T}}. \tag{19}$$

Note from (19) that: (i) the zeroed coefficients of the sparse filter are recreated, which are indicated by boxes; and (ii) new coefficients appear, the underlined ones, as a consequence of the border effect. As a second example, for $L=3$ and $N=7$, one obtains in $N_{\mathrm{s}}=3$, $M=5$, $\mathbf{i} = [i_0\ i_1\ i_2\ i_3\ i_4]^{\mathrm{T}}$, $\mathbf{w}_{\mathrm{s}} = [w_0\ \ 0\ \ 0\ \ w_3\ \ 0\ \ 0\ \ w_6]^{\mathrm{T}}$. Then, the equivalent coefficient vector is

$$\mathbf{w}_{\mathrm{i}} = \Big[\underline{i_0w_0}\ \ \underline{i_1w_0}\ \ i_2w_0\ \ \boxed{i_3w_0 + i_0w_3}\ \ \boxed{i_4w_0 + i_1w_3}\ \ i_2w_3\ \ \boxed{i_3w_3 + i_0w_6}\ \ \boxed{i_4w_3 + i_1w_6}\ \ i_2w_6\ \ \underline{i_3w_6}\ \ \underline{i_4w_6}\Big]^{\mathrm{T}} \tag{20}$$

where the recreated coefficients are indicated by boxes, while the underlined ones arise from the border effect.

3. GENERALIZED BORDER EFFECT REMOVAL FOR IFIR FILTERS

The border effect, described in Section 2, leads to an important performance loss in several applications [10], [11]. In [11], a procedure for removing such an effect in IFIR structures considering input interpolators is described, resulting in the removed border effect IFIR (RBEIFIR) filter. This procedure is presented there for a particular case $(L=2)$ and is based on a transformation matrix $\mathbf{T}$, applied to the equivalent coefficient vector, given by [12]

$$\mathbf{w}'_{\mathrm{i}} = \mathbf{T}\mathbf{w}_{\mathrm{i}} = \mathbf{T}\mathbf{I}\mathbf{w}_{\mathrm{s}} = \mathbf{T}\mathbf{W}_{\mathrm{s}}\mathbf{i} \tag{21}$$

where $\mathbf{w}'_{\mathrm{i}}$ is the equivalent coefficient vector with the removed border effect.

In this section, the aforementioned procedure is generalized for different values of L and for the structure having an output interpolator. Thus, by taking into account that the first $L-1$ and last $L-1$ coefficients of the equivalent coefficient vector arises from the border effect [see (18), (19) and (20)], the following generalized transformation matrix, with dimensions $N\times(N+M-1)$, can be written as:

$$\mathbf{T} = \begin{bmatrix} \underbrace{\begin{matrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{matrix}}_{L-1\text{ columns}} & \underbrace{\begin{matrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{matrix}}_{N\text{ columns}} & \underbrace{\begin{matrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{matrix}}_{L-1\text{ columns}} \end{bmatrix}. \tag{22}$$

By using (21) and (22), the equivalent coefficient vector with the removed border effect is obtained. Then, considering (20) for the previous example, one obtains

$$\mathbf{w}'_{\mathrm{i}} = \Big[i_2w_0\ \ \boxed{i_3w_0 + i_0w_3}\ \ \boxed{i_4w_0 + i_1w_3}\ \ i_2w_3\ \ \boxed{i_3w_3 + i_0w_6}\ \ \boxed{i_4w_3 + i_1w_6}\ \ i_2w_6\Big]^{\mathrm{T}}. \tag{23}$$

The border effect removal procedure is carried out by replacing the input vector (of the output block of the IFIR structure) by a new modified one. For the case of an input-interpolator-IFIR structure, the input vector of the second filter (sparse), given by (7), is replaced by

$$\tilde{\mathbf{x}}'(n) = \mathbf{I}^{\mathrm{T}}\mathbf{T}^{\mathrm{T}}\mathbf{x}(n) = \mathbf{I}'^{\mathrm{T}}\mathbf{x}(n) \tag{24}$$

with

$$\mathbf{x}(n) = \left[x(n)\ \ x(n-1)\ \ \cdots\ \ x(n-N+1)\right]^{\mathrm{T}}. \tag{25}$$

For the case of the output-interpolator-IFIR structure, the vector to be replaced is the input vector of the interpolator filter. Thus, (9) is replaced by

$$\hat{\mathbf{x}}'(n) = \mathbf{W}_s^{\mathrm{T}}\mathbf{T}^{\mathrm{T}}\mathbf{x}(n) \,. \tag{26}$$

The computational cost for implementing the border effect removal procedure is small [11]. For instance, by considering an input-interpolator-IFIR structure, $L=2$, $M=3$, $N=5$, and $\mathbf{i}=[i_0 \;\; i_1 \;\; i_2]^{\mathrm{T}}$, the modified input vector is given by

$$\tilde{\mathbf{x}}'(n) = \mathbf{I}^{\mathrm{T}}\mathbf{T}^{\mathrm{T}}\mathbf{x}(n) = \begin{bmatrix} i_1 x(n) + i_2 x(n-1) \\ \mathbf{i}^{\mathrm{T}}\mathbf{x}_{\mathrm{m}}(n) \\ \mathbf{i}^{\mathrm{T}}\mathbf{x}_{\mathrm{m}}(n-1) \\ \mathbf{i}^{\mathrm{T}}\mathbf{x}_{\mathrm{m}}(n-2) \\ i_0 x(n-3) + i_1 x(n-4) \end{bmatrix}^{\mathrm{T}} \tag{27}$$

with

$$\mathbf{x}_{\mathrm{m}}(n) = [x(n) \;\; x(n-1) \;\; \cdots \;\; x(n-M+1)]^{\mathrm{T}} \,. \tag{28}$$

To obtain (27), at each iteration, the following operations are required:

i) Two multiplications and one sum to obtain the first element of the current iteration.
ii) One multiplication and one sum to obtain the second element of the current iteration from the first element of the previous iteration.
iii) One multiplication and one subtraction to obtain the last element of the current iteration from the last but one element of the previous iteration.

Therefore, for the case considered, 4 multiplications and 3 sums are required for computing (27); meanwhile, the standard IFIR structure requires 3 multiplications and 2 sums. In general, $M+L-1$ multiplications and $M+L-2$ sums are required per iteration for computing (27). By comparing (27) with (7) $2L-2$ more operations are necessary. Since L is small, the increase in computational burden becomes negligible.

4. FULLY ADAPTIVE IFIR FILTERS WITH REMOVED BORDER EFFECT

The use of a border effect removal procedure in AIFIR filters is straightforward, resulting in the adaptive removed border effect IFIR (ARBEIFIR) filter [11]. Since the use of an adaptive interpolator can improve the structure performance, in this section, the fully adaptive IFIR (FAIFIR) structure is derived [7]-[9]. The disadvantages of such an approach are related to the cascaded structure adaptation process, as described in [12].

The update expression for the fully adaptive RBEIFIR (FARBEIFIR) structure is more complex than for the ARBEIFIR one. This is due to the fact that the FARBEIFIR filter is a cascaded adaptive structure and the direct application of the LMS expressions is not possible. Thus, a similar approach as in [12] must be adopted. By using (16), (24), and (25) the error signal is given by [12]

$$\begin{aligned} e(n) &= d(n) - \mathbf{w}_s^{\mathrm{T}}(n)\mathbf{I}^{\mathrm{T}}(n)\mathbf{T}^{\mathrm{T}}\mathbf{x}(n) \\ &= d(n) - \mathbf{i}^{\mathrm{T}}(n)\mathbf{W}_s^{\mathrm{T}}(n)\mathbf{T}^{\mathrm{T}}\mathbf{x}(n). \end{aligned} \tag{29}$$

Note that in (29), the vectors and matrices depending on the coefficients of the IFIR filter are now time varying. Similarly to [12] and considering (29), the LMS update expression for the sparse filter coefficients is written as

$$\mathbf{w}_{\mathrm{s}}(n+1) = \mathbf{w}_{\mathrm{s}}(n) + 2\mu_1 e(n)\mathbf{I}^{\mathrm{T}}(n)\mathbf{T}^{\mathrm{T}}\mathbf{x}(n) \tag{30}$$

and for the interpolator, as

$$\mathbf{i}(n+1) = \mathbf{i}(n) + 2\mu_2 e(n)\mathbf{W}_s^{\mathrm{T}}(n)\mathbf{T}^{\mathrm{T}}\mathbf{x}(n) \,. \tag{31}$$

The implementation of (30) and (31) involves matrix-vector products, resulting in a high computational cost [12]. However, the same simplifying approximations adopted in [12] can also be used here. Then, the update expressions for the LMS FARBEIFIR structure becomes

$$\mathbf{w}_{\mathrm{s}}(n+1) = \mathbf{w}_{\mathrm{s}}(n) + 2\mu_1 e(n)\tilde{\mathbf{x}}'(n) \tag{32}$$

and

$$\mathbf{i}(n+1) = \mathbf{i}(n) + 2\mu_2 e(n)\hat{\mathbf{x}}'(n) \,. \tag{33}$$

It is important to highlight that (32) and (33) are approximate expressions and a special care must be taken for the choice of the step-size and control parameters [12]. Expressions for the normalized LMS (NLMS) algorithm are obtained analogously to (32) and (33), considering the procedure presented in [12]. Such expressions are valid for adapting both input- and output-interpolator-IFIR structures. By using the output-interpolator-IFIR structure a lower computational burden is obtained. Such a characteristic is also observed in the implementation of the standard FAIFIR structure.

5. SIMULATION RESULTS

In this section, numerical simulations are presented aiming to assess the performance of the FARBEIFIR structure, as compared with other implementations of adaptive IFIR filters (AIFIR, FAIFIR and ARBEIFIR). The used examples refer to a system identification problem [14]. The performance is assessed in terms of the MSE characteristic, obtained from Monte Carlo simulations (average of 100 runs). The interpolation factor for all examples is $L=2$ and the implementations considering fixed interpolators (AIFIR and ARBEIFIR) use a linear interpolator given by $\mathbf{i}=[0{,}5 \;\; 1 \;\; 0{,}5]^{\mathrm{T}}$. The input signal is white and Gaussian with variance $\sigma_x^2 = 1$ (similar results in terms of the steady-state MSE value are obtained for colored inputs). An additive Gaussian noise with variance $\sigma_z^2 = 10^{-8}$ is added to the output of the plant ($SNR = 80\,\mathrm{dB}$).

Example 1: In this example, the plant impulse response $w_{\mathrm{p}1}(n)$ is illustrated in Fig. 3. The memory size of the adaptive filters is $N=71$ and they are adapted using the NLMS algorithm with $\alpha_1 = 0.5$, $\alpha_2 = 0.05$ and $\psi_1 = \psi_2 = 10^{-8}$ [12]. The obtained results are shown in Fig. 4. One observes a better performance of the FARBEIFIR structure (note that its steady-state MSE value is very close to the noise variance) in comparison with the other ones.

Example 2: In this example, the coefficients of the plant $w_{\mathrm{p}2}(n)$ are illustrated in Fig. 5. One can verify here that the impact of the border effect is smaller due to the fact that the first and the last coefficients of the plant are equal to zero [11]. The memory size of the adaptive filters is $N=101$, adapted using the LMS algorithm with $\mu_1 = \mu_2 = 0.1\mu_{\max}$, where $\mu_{\max}$ is the upper stability bound for the step-size parameter (experimentally determined). In Fig. 6, the obtained MSE curves are depicted. Now the performance of the different IFIR structures are very close to each other, noting a better performance of the FARBEIFIR structure. Such a fact corroborates the robustness of the proposed implementation, presenting the best performance among the IFIR structures even if the border effect does not occur.

6. CONCLUSIONS

In this paper, a new procedure for implementing fully adaptive IFIR structures with the removed border effect is presented. In comparison with the other implementations of adaptive IFIR structures available in the literature, the proposed approach presents a considerable improvement in the MSE behavior with a slight computational cost increment.

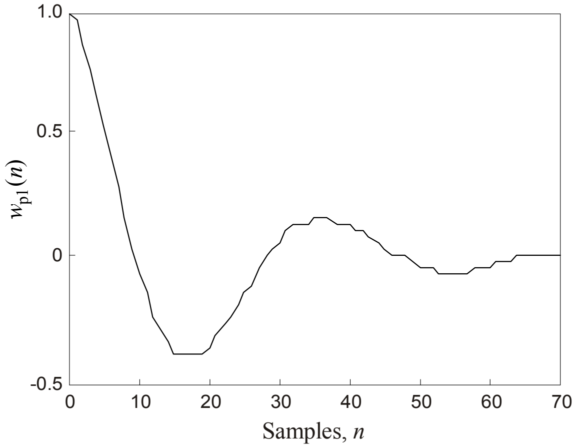

Fig. 3. Example 1. Plant impulse response.

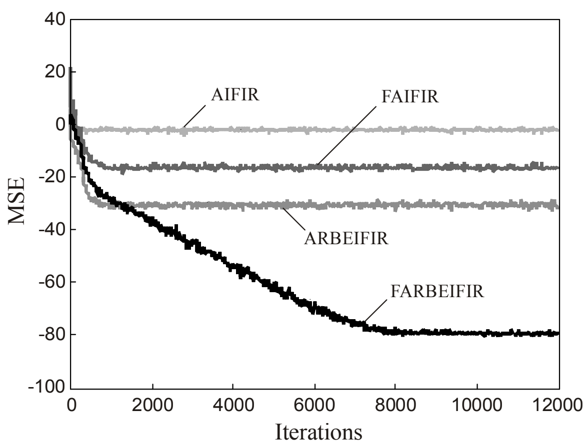

Fig. 4. Example 1. MSE curves.

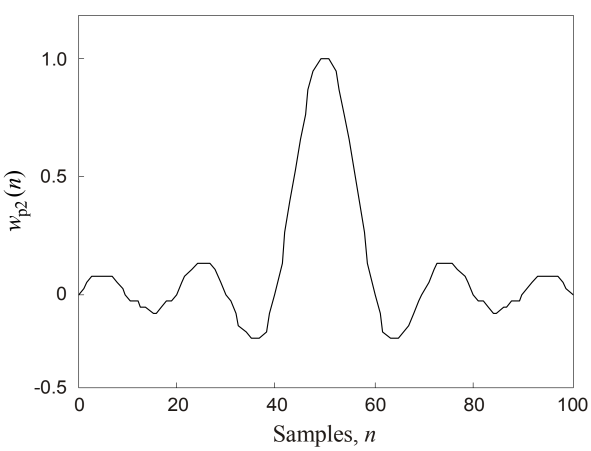

Fig. 5. Example 2. Plant impulse response.

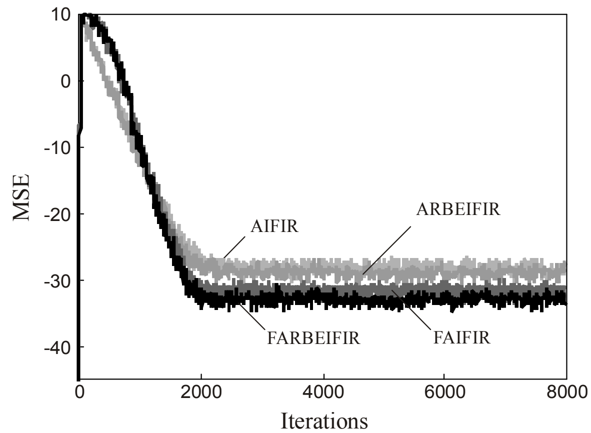

Fig. 6. Example 2. MSE curves.

7. REFERENCES

[1] 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.

[2] A. Abousaada, T. Aboulnasr, and W. Steenaart, “An echo tail canceller based on adaptive interpolated FIR filtering,” *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 39, no. 7, pp. 409-416, Jul. 1992.

[3] O. J. Tobias, R. Seara Jr., and R. Seara, “Echo canceller based on adaptive interpolated FIR filters,” in *Proc. IEEE Int. Telecom. Symp.*, Natal, Brazil, Sept. 2002, pp. 1-5.

[4] 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.

[5] S. Kuo and D. R. Morgan, *Active Noise Control Systems*, John Wiley & Sons, 1996.

[6] L. 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.

[7] M. D. Grosen, “New FIR structures for fixed and adaptive digital filters,” Ph.D. Dissertation, University of California, Santa Barbara, CA, United States, 1987.

[8] 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.

[9] 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.

[10] E. L. O. Batista, O. J. Tobias, and R. Seara, “A mathematical framework to describe interpolated adaptive Volterra filters,” in *Proc. IEEE Int. Telecomm. Symp.*, Fortaleza, Brazil, Sept. 2006, pp. 144-149.

[11] E. L. O. Batista, O. J. Tobias, and R. Seara, “Border effect removal for IFIR and interpolated Volterra filters,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Honolulu, USA, vol. 3, Apr. 2007, pp. 1329-1332.

[12] E. L. O. Batista, O. J. Tobias, and R. Seara, “New insights in adaptive cascaded FIR structure: application to fully adaptive interpolated FIR structures,” in *Proc. 15th European Signal Processing Conf. (EUSIPCO)*, Poznan, Poland, vol. 1, Sep. 2007, pp. 370-374.

[13] R. Seara, J. C. M. Bermudez, and E. Beck, “A new technique for the implementation of adaptive IFIR filters,” in *Proc. Int. Symp. Signals, Systems, Electronics (ISSSE)*, Paris, France, vol. 2, Sept. 1992, pp. 644-647.

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