ON ADAPTIVE INTERPOLATED FIR FILTERS

Radu Ciprian Bilcu[1], *Pauli Kuosmanen*[2], *Karen Egiazarian*[3]

Tampere University of Technology
Institute of Signal Processing
Korkeakoulunkatu 1, FIN-33720, Tampere, Finland
[1]bilcu@cs.tut.fi, [2]pqo@cs.tut.fi, [3]karen@cs.tut.fi

ABSTRACT

The computational complexity and memory load of the Adaptive Finite Impulse Response (AFIR) filter are significant for a large filter size. Adaptive Interpolated FIR (AIFIR) filter, which uses a sparse adaptive filter followed by an interpolator, has been shown to be a better alternative. However, when the AIFIR filter is implemented, the coefficients of the interpolator must be designed in advance based on prior information about the application at hand. Such information is not always available and the design of a proper interpolator is sometimes difficult. In this paper we introduce a new structure called Double Adaptive Interpolated FIR (DAIFIR) filter in which the fixed interpolator is replaced by an adaptive filter of the same length. We show by means of simulations that the behavior of the proposed structure is close to the behavior of the AIFIR having a proper designed interpolator. In situations in which a fixed interpolator cannot be designed in advance the DAIFIR might be a good alternative.

1. INTRODUCTION

The Adaptive Finite Impulse Response (AFIR) filters are widely used in many practical applications due to their advantages, such as, simplicity of implementation [1], [2]. However, their computational complexity and memory load is proportional to the number of coefficients of the adaptive filter. As a consequence, in applications where a long adaptive filter must be implemented, the computational complexity and memory load can be huge (for example, in echo cancellation, there is a necessity to use a large FIR adaptive filter to model the echo path which highly increase the complexity, [3]). In such applications, the Adaptive Interpolated FIR (AIFIR) filters represents an interesting alternative which gives an important reduction of the arithmetic operations for both filtering and weight updating and also improves the memory usage.

The class of AIFIR filters was derived from the class of Interpolated FIR (IFIR) filters first introduced by Neuvo et all. in [4]. The main idea of the IFIR filters is to remove a number of coefficients from an FIR filter and then recreate them using an interpolator. In [4], the sparse filter and the interpolator have fixed coefficients and they are designed based on some priory information about the application at hand [5], [6], [7].

The class of AIFIR was introduced to reduce the computational complexity and memory load in applications in where a large adaptive filter is to be used (see [3], [8], [9] and the references therein). In the case of AIFIR, a sparse adaptive filter followed by an interpolator with fixed coefficients is used, instead of a long adaptive filter, which highly decrease the complexity. We note, that the interpolator plays an important role in the performance of the AIFIR and design procedures for the interpolator with fixed coefficients can be found in [4], [6]. In some applications, no information about the Wiener solution is available and the design of a proper interpolator might be difficult.

In this paper, we introduce a new structure called Double Adaptive FIR (DAIFIR) filter in which the fixed interpolator is replaced by an adaptive filter of the same length. We show by means of simulations that the behavior of the proposed structure is close to the behavior of the AIFIR having a proper designed interpolator. In situations in which the interpolator cannot be designed in advance, the DAIFIR might be a good alternative. Comparison between the AFIR, AIFIR and DAIFIR filters in terms of computational complexity and memory load are also presented.

This work is protected under the Finnish patent application number: 20035050.

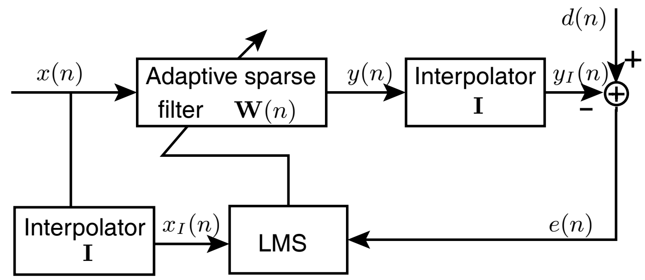

Fig. 1. Block diagram of the Adaptive Interpolated FIR filter.

2. THE ADAPTIVE INTERPOLATED FIR FILTER

The block diagram of an AIFIR filter is presented in Fig. 1, where $\mathbf{W}(n)$ represents a sparse FIR adaptive filter having L zeros between nonzero coefficients, the block denoted by $\mathbf{I}$ represents the interpolation filter with fixed coefficients which recreates the removed samples from $\mathbf{W}(n)$, $x(n)$ is the input signal, $d(n)$ is the desired signal and $e(n)$ is the output error.

The coefficients of the adaptive sparse filter $\mathbf{W}(n)$ are adapted, such that the expected value of the squared error is minimized. To handle the sparse nature of the filter $\mathbf{W}(n)$ a constrained approach can be used (see [8], [10]) and the constrained cost function to be minimized is the following:

$$\begin{aligned} &\text{minimize} \quad E\left[e^2(n)\right], &(1)\\ &\text{subject to} \quad \mathbf{C}^t\mathbf{W}(n) = \mathbf{f}. &(2)\end{aligned}$$

($\mathbf{C}$ and $\mathbf{f}$ given below in (9) and (10))

From (1) and (2) the adaptive constrained LMS algorithm, to adapt the sparse FIR filter $\mathbf{W}(n)$ is derived and it can be described by the following steps:

1. Compute the output of the filter $\mathbf{W}(n)$:

$$y(n) = \mathbf{W}^t(n)\mathbf{X}(n), \tag{3}$$

where

$$\mathbf{X}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^t, \tag{4}$$

is the vector of the past N samples from the input signal $x(n)$ and N is the length of the adaptive filter $\mathbf{W}(n)$.

2. Compute the output of the interpolator:

$$y_I(n) = \mathbf{I}^t\mathbf{Y}(n), \tag{5}$$

where $\mathbf{I} = [i_1, i_2, \ldots, i_M]^t$ is the vector containing the interpolator coefficients and $\mathbf{Y}(n) = [y(n), y(n-1), \ldots, y(n-M+1)]^t$ is the vector of the past M samples from the signal $y(n)$.

3. Compute the output error:

$$e(n) = d(n) - y_I(n), \tag{6}$$

4. Compute the filtered input vector $\mathbf{X}_I(n)$ (see Fig. 1):

$$\mathbf{X}_I(n) = \sum_{j=0}^{M-1} i_j\mathbf{X}(n-j), \tag{7}$$

where $\mathbf{X}(n)$ is given in (4).

5. Update the sparse adaptive filter weights:

$$\mathbf{W}(n+1) = \mathbf{F}\left\{\mathbf{W}(n) + \mu e(n)\mathbf{X}_I(n)\right\} + \mathbf{q}, \tag{8}$$

where $\mathbf{F} = \mathbf{Id} - \mathbf{C}\left(\mathbf{C}^t\mathbf{C}\right)^{-1}\mathbf{C}^t$ is the projection matrix ($\mathbf{Id}$ being the identity matrix of order N) and $\mathbf{q} = \mathbf{C}\left(\mathbf{C}^t\mathbf{C}\right)^{-1}\mathbf{f}$ is the correction vector (see [10]).

The matrix $\mathbf{C}$ and the vector $\mathbf{f}$ for N odd and $L = 1$ are expressed as follows:

$$\mathbf{C}^t = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 1 & 0 \end{bmatrix}_{K\times N} \tag{9}$$

$$\mathbf{f} = [0\ldots0]^t_{1\times K} = \mathbf{0}_{K\times1} \tag{10}$$

where K is the number of non-zero coefficients in the sparse filter $\mathbf{W}(n)$.

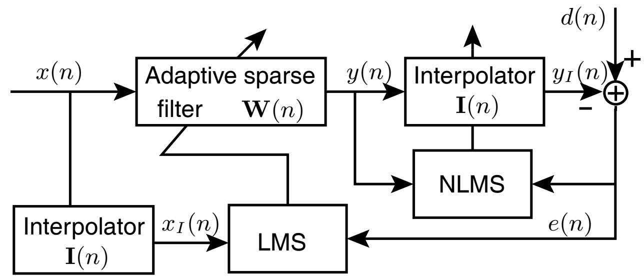

Fig. 2. Block diagram of the proposed filter.

As a consequence, the matrix $\mathbf{F}$ and the vector $\mathbf{q}$ in (8) can be written as follows:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 1 \end{bmatrix}_{N\times N} \tag{11}$$

$$\mathbf{q} = [0\ldots0]^t_{1\times N} \tag{12}$$

From (11) and (12), one can see that (8) is equivalent to the update equation of the standard LMS, in which just K coefficients are adapted provided that $\mathbf{W}(n)$ is initialized with zeros. As a consequence, the multiplication by $\mathbf{F}$ and addition of $\mathbf{q}$ does not introduce extra computations in (8).

For detailed analysis of the AIFIR see [8] and [11]. In the referred papers, the behavior of the mean weight vector and the behavior of the weight-error covariance matrix are studied. However, in the papers published so far, the interpolator has fixed coefficients and it is designed based on some available information about the optimum solution.

In this paper, the AIFIR algorithm is modified such that the coefficients of the interpolator are adapted using the Normalized LMS (NLMS) algorithm.

3. THE NEW APPROACH

The block diagram for system identification of the new DAIFIR is depicted in Fig. 2, where we have used the same notations as in Fig. 1. The difference between the two approaches is that in the new implementation the interpolator (denoted by $\mathbf{I}(n)$) has time-variable coefficients adapted by the NLMS algorithm.

The new algorithm can be mainly described by the same five steps given in the previous section. The difference is that, after the fifth step the coefficients of the interpolator $\mathbf{I}(n)$ are also adapted as follows:

$$\mathbf{I}(n+1) = \mathbf{I}(n) + \frac{\mu_I}{\epsilon + \sum_{j=0}^{M-1} y^2(n-j)} e(n)\mathbf{Y}(n). \tag{13}$$

where μ_I is the step-size used to update the coefficients of the interpolator, $e(n)$ is the output error (see Fig. 2), $\mathbf{I}(n) = [i_1(n), \ldots, i_M(n)]^t$ is the $M \times 1$ vector containing the coefficients of the interpolator, $\mathbf{Y}(n) = [y(n), \ldots, y(n-M+1)]^t$

is the vector containing the past M samples from the signal $y(n)$ and ϵ is a small constant.

The DAIFIR filter has a slightly increased complexity compared with the AIFIR due to the extra computations required by the interpolator adaptation as shown in Table 1. When compared with the AFIR filter, the AIFIR and the proposed DAIFIR have lower complexity (especially the memory load).

Table 1. Complexity in terms of memory load and number of arithmetic operations for the AFIR, AIFIR and DAIFIR filters (the numerical values are those used in the experiments).

Ad. filter	Mem. load	Mult.+Div.	Add.+Sub.
AFIR	2N+4=258	2N+1=255	2N=254
AIFIR	3M+N+3+ +2K=195	K(M+1)+ +2K+1=229	K(M+1)+ M-1=198
DAIFIR	3M+N+6+ +2K=198	K(M+2)+ +3M+3=239	K(M+1)-1 +3M=216

4. SIMULATIONS AND RESULTS

The proposed DAIFIR filter is compared with AIFIR and AFIR filters, in system identification framework where the desired signal $d(n)$ is obtained from an FIR filter $\mathbf{W}$ of length N as follows:

$$d(n) = \sum_{i=1}^{N} w_i x(n-i+1) + v(n).$$

where w_i is the i^{th} coefficient of $\mathbf{W}$ and $v(n)$ is the output noise.

The output signal to noise ratio, in all experiments, was $SNR = 30dB$. The length of the FIR filter $\mathbf{W}$, the lengths of the sparse filters in AIFIR and DAIFIR and the length of the AFIR filter was $N = 127$. The sparse filters in the AIFIR and DAIFIR have just $K = 19$ nonzero coefficients (there are $L = 6$ zeros between nonzero coefficients). The length of the interpolating filters $\mathbf{I}$ and $\mathbf{I}(n)$ in AIFIR and DAIFIR was $M = 9$. The complexity of the compared adaptive filters, in terms of number of mathematical operations and memory load, are shown in Table 1. We can see that the DAIFIR and AIFIR has comparable complexities which are lower than the complexity of the AFIR filter.

Two experiments were done in order to compare the behavior of the AFIR, AIFIR and DAIFIR filters. In the first experiment, the unknown system has a low-pass frequency response with normalized cut-off frequency $f_c = 0.05$. The interpolator $\mathbf{I}$ of the AIFIR was designed to have also a low-pass frequency response with the same normalized cut-off frequency. In the second experiment, the unknown system has a high-pass frequency response with the normalized cut-off frequency $f_c = 0.95$, while the interpolator of the AIFIR was the same as in the first experiment.

The output mean squared error (MSE) of each of the compared filters are shown in Fig. 3, Fig. 4 and Fig. 5 for the first experiment and in Fig. 6, Fig. 7 and Fig. 8, for the second experiment. The learning curves shown in the figures were obtained by averaging 100 Monte-Carlo simulations of 10^4 iterations each. From Fig. 7 and Fig. 8, we can see that the proposed filter performs better than the AIFIR in the second experiment. This is because the AIFIR uses a fixed interpolator which was not properly designed, while the DAIFIR uses an adaptive interpolator.

5. CONCLUSIONS

In this paper, a modification of the known AIFIR filter is proposed in which the interpolator used for reconstruction of the zero taps is adapted using the NLMS algorithm. The new DAIFIR filter can be used in the applications where no information about the optimal solution is available, and, therefore a fixed interpolator cannot be designed.

6. REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[2] M. G. Bellanger, *Adaptive Digital Filters and Signal Analysis*, Marcel Dekker Inc., New-York and Basel, 1987.

[3] A. Abousaada, T. Aboulnasr, and W. Steenart, “An Echo Tail Canceller Based on Adaptive Interpolated FIR Filtering,” *in IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 39, no. 7, pp. 409–416, July 1992.

[4] Y. Neuvo, C-Y. Dong, and S. K. Mitra, “Interpolated Finite Impulse Response Digital Filters,” *in IEEE Transactions on Acoustic, Speech Signal Processing*, vol. ASSP-32, no. 3, pp. 563–570, June 1984.

[5] T. Saramaki, Y. Neuvo, and S. K. Mitra, “Design of Computationally Efficient Interpolated FIR Filters,” *in IEEE Transactions on Circuits and Systems*, vol. 35, no. 1, pp. 70–88, Jan. 1988.

[6] O. Vainio, “Design and Efficient Implementations of Predictive FIR Filters,” *in IEEE Transactions on Instrumentation and Measurement*, vol. 44, no. 4, pp. 864–868, Aug. 1995.

[7] Y. Neuvo, G. Rajan, and S. K. Mitra, “Design of Narrow-Band FIR Bandpass Digital Filters with Reduced Arithmetic Complexity,” *in IEEE Transactions on Circuits and Systems*, vol. 34, no. 4, pp. 409–419, Apr. 1987.

[8] 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,” *in IEE Proc.-Vis. Image Signal Processing*, vol. 148, no. 5, pp. 337–347, Oct. 2001.

[9] L. S. Resende, C. A. F. Rocha, and M. G. Bellanger, “A New Structure for Adaptive IFIR Filtering,” in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Istanbul, Turkey, June 2000, pp. 400–403.

[10] O. L. Frost, “An Algorithm for Linearly Constrained Adaptive Array Processing,” *in Proceedings of the IEEE*, vol. 60, no. 8, pp. 926–935, Aug. 1972.

[11] O. J. Tobias, J. C. M. Bermudez, and N. Bershad, “Mean Weight Behavior of the Filtered X-LMS Algorithm,” *in IEEE Transactions on Signal Processing*, vol. 48, no. 4, pp. 1061–1075, Apr. 2000.

[12] R. Manduchi, “Two-Dimensional IFIR Structures Using Generalized Factorizable Filters,” *in IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 7, pp. 564–576, July 1997.

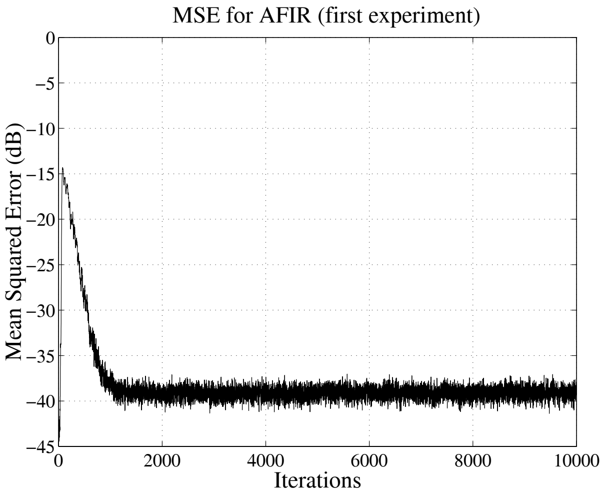

Fig. 3. Mean squared error of the AFIR (first experiment)

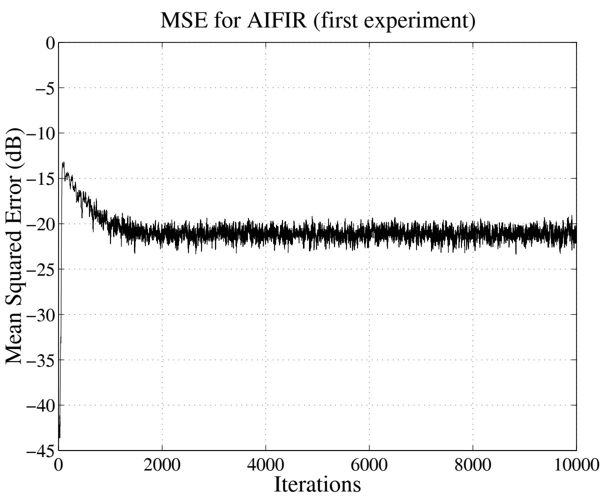

Fig. 4. Mean squared error of the AIFIR (first experiment)

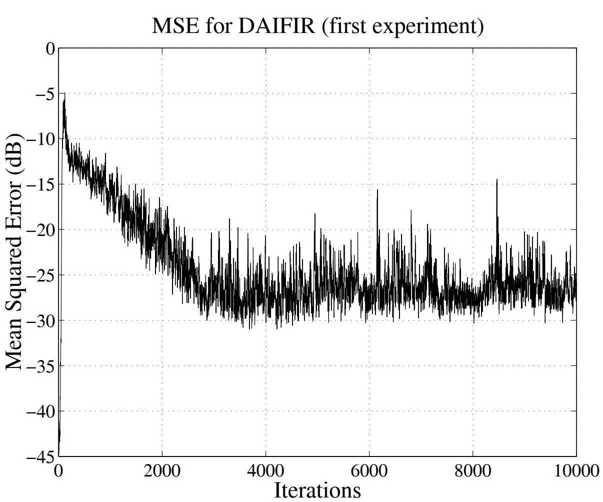

Fig. 5. Mean squared error of the DAIFIR (first experiment)

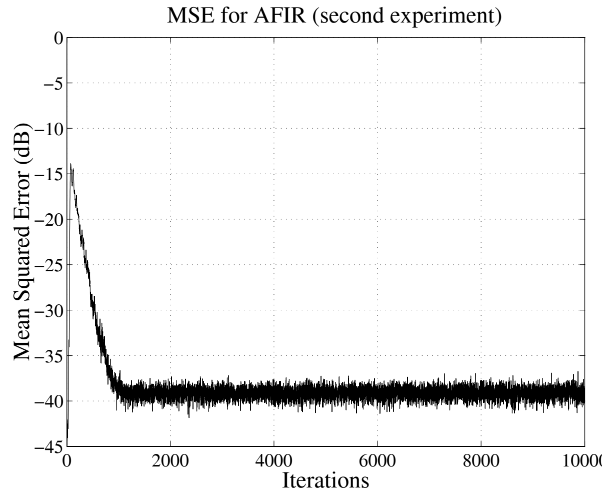

Fig. 6. Mean squared error of the AFIR (second experiment)

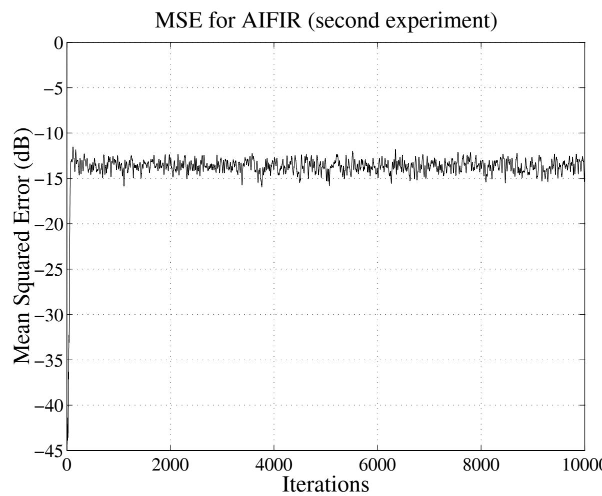

Fig. 7. Mean squared error of the AIFIR (second experiment)

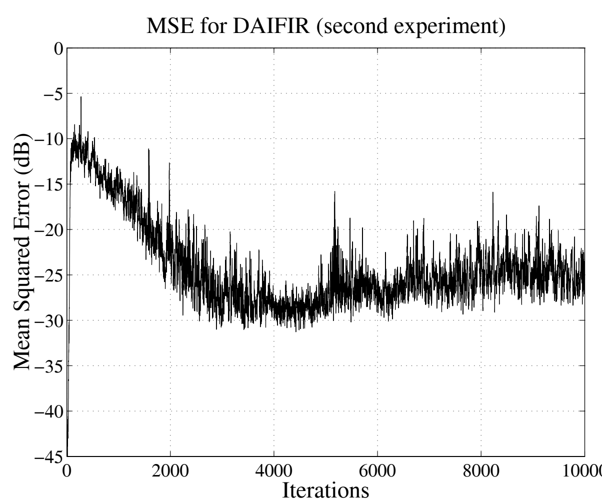

Fig. 8. Mean squared error of the DAIFIR (second experiment)