IMPROVING THE CONVERGENCE OF ADAPTIVE HAMMERSTEIN FILTERS

Eduardo L. O. Batista — Department of Informatics and Statistics
Rui Seara — Department of Electrical Engineering
LINSE – Circuits and Signal Processing Lab., Federal University of Santa Catarina, 88040-900, Florianópolis, SC, Brazil
E-mails: ebatista@ieee.org, seara@linse.ufsc.br

ABSTRACT

The implementation of adaptive Hammerstein filters involves updating the coefficients of two cascaded blocks, namely, a memoryless nonlinearity and a linear filter. Such an update process presents important numerical problems mainly due to the non-uniqueness of the coefficient values that lead to optimum performance. These problems can be circumvented by keeping constant (not adapting) one of the filter coefficients, which however may significantly slow down the convergence of the adaptive algorithm. In this context, this paper presents a novel approach to implement adaptive Hammerstein filters in which a coefficient normalization strategy is used to overcome the aforementioned numerical problems. Thus, enhanced convergence speed is obtained with a small increase in the computational burden. Simulation results are presented to corroborate the effectiveness of the proposed strategy.

Index Terms—Adaptive filters, Hammerstein filters, NLMS algorithm, nonlinear filters.

1. INTRODUCTION

Signal processing systems involving adaptive filtering algorithms have been extensively used over the last decades. This is in part due to the processing power increase of the modern digital signal processors (DSPs), which has allowed practical implementations of sophisticated adaptive filtering structures. The nonlinear adaptive filters [1] are very good examples of such structures.

Nonlinear filters and their adaptive implementations are usually classified according to their underlying nonlinear functions due to the unavailability of a unifying nonlinear filtering theory. In this context, one important class is that of the Hammerstein filters [2], which are composed of a memoryless nonlinearity cascaded with a linear filter. As described in [2], such structures are appropriate for modeling many real-world systems and, as a consequence, the Hammerstein filters have been used in several practical applications. In particular, their adaptive implementations are used in various acoustic echo cancellation systems [3]-[5].

In adaptive applications, the update of the coefficients of the two blocks (memoryless nonlinearity and FIR filter) that compose a Hammerstein filter may present significant numerical problems. This is due to the fact that optimum performance can be obtained with one of such blocks having very small coefficient values whereas the coefficients of the other have large values. Since small coefficient values may result in loss of numerical precision and large values in arithmetic overflow, the adaptive algorithm may fail in its purpose of searching for optimum coefficient values. As described in [2], such numerical issues can be circumvented by keeping constant (not adapting) the value of one of the filter coefficients. This strategy, in spite of resulting in a unique solution that contributes to the stability of the adaptive algorithm [6], often produces a significant degradation on the convergence speed. To overcome this problem, this paper presents a novel approach for implementing adaptive Hammerstein filters based on the use of a coefficient normalization strategy to avoid the aforementioned numerical issues. Such a strategy can be used with different adaptive algorithms despite the fact that the algorithm of choice here is the normalized least-mean-square (NLMS) because of its very good tradeoff between computational complexity and performance.

This paper is organized as follows. In Section 2, the implementation of adaptive Hammerstein filters using the NLMS algorithm is discussed. Section 3 starts with a discussion regarding the implementation obtained by keeping constant one of the coefficients and then describes the proposed implementation strategy, its numerical characteristics and the required computational burden. Results of numerical simulations are presented in Section 4 aiming to assess the performance of Hammerstein implementations obtained by using the proposed strategy. Concluding remarks are presented in Section 5.

2. NLMS HAMMERSTEIN FILTERS

Implementations of Hammerstein filters are obtained by using different types of memoryless nonlinearities and of linear filters [2]-[5]. Here, we consider the implementation from [3], which is composed of a polynomial memoryless nonlinearity $\mathbf{p}$ followed by a linear FIR filter $\mathbf{w}$ (see Fig. 1). It is important to note that, despite considering such implementation, the strategy developed here may be easily extended to other Hammerstein implementations.

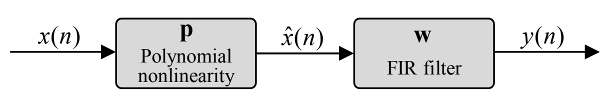

Fig. 1. Block diagram of a typical Hammerstein filter.

In Fig. 1, $x(n)$ represents the input signal, $y(n)$, the output signal, and $\hat{x}(n)$, the input signal for $\mathbf{w}$. The input-output relationship for $\mathbf{p}$ is

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

$$\hat{x}(n) = p_1 x(n) + p_2 x^2(n) + \cdots + p_M x^M(n) \quad (1)$$

with M representing the nonlinearity order and p_m, the mth order coefficient. By defining an input vector as

$$\mathbf{x}_\mathrm{p}(n) = [x(n) \;\; x^2(n) \;\; x^3(n) \;\; \cdots \;\; x^M(n)]^\mathrm{T} \quad (2)$$

and a corresponding coefficient vector as

$$\mathbf{p} = [p_1 \;\; p_2 \;\; p_3 \;\; \cdots \;\; p_M]^\mathrm{T} \quad (3)$$

expression (1) can be rewritten as

$$\hat{x}(n) = \mathbf{p}^\mathrm{T}\mathbf{x}_\mathrm{p}(n). \quad (4)$$

Moreover, defining the coefficient vector of the FIR filter as

$$\mathbf{w} = [w_0 \;\; w_1 \;\; w_2 \;\; \cdots \;\; w_{N-1}]^\mathrm{T} \quad (5)$$

its input-output relationship can be written as

$$y(n) = \mathbf{w}^\mathrm{T}\hat{\mathbf{x}}(n) \quad (6)$$

with $\hat{\mathbf{x}}(n) = [\hat{x}(n) \;\; \hat{x}(n-1) \;\; \cdots \;\; \hat{x}(n-N+1)]^\mathrm{T}$ representing the input vector and N, the memory size. Aiming to obtain an input-output relationship for the whole structure of the Hammerstein filter, one can define an input matrix as

$$\mathbf{X}(n) = [\mathbf{x}_\mathrm{p}(n) \;\; \mathbf{x}_\mathrm{p}(n-1) \;\; \cdots \;\; \mathbf{x}_\mathrm{p}(n-N+1)] \quad (7)$$

and, considering (6) and that $\hat{\mathbf{x}}(n) = \mathbf{X}^\mathrm{T}(n)\mathbf{p}$, the following input-output relationship is obtained:

$$y(n) = \mathbf{w}^\mathrm{T}\mathbf{X}^\mathrm{T}(n)\mathbf{p}. \quad (8)$$

2.1. Update Equations

The expression to update the coefficients of the memoryless nonlinearity $\mathbf{p}$ using the NLMS algorithm is obtained by minimizing the squared Euclidean norm of

$$\delta\mathbf{p}(n+1) = \mathbf{p}(n+1) - \mathbf{p}(n) \quad (9)$$

subject to

$$\mathbf{p}^\mathrm{T}(n+1)\mathbf{X}(n)\mathbf{w}(n) = d(n) \quad (10)$$

with $\mathbf{p}(n)$ and $\mathbf{w}(n)$ representing the adaptive versions of (3) and (5), respectively, $\mathbf{p}(n+1)$, the *a posteriori* version of $\mathbf{p}(n)$ [7], and $d(n)$, the desired signal. Then, solving the minimization problem described by (9) and (10) (using an approach analogous to the one presented in [8]), the following update expression is obtained:

$$\mathbf{p}(n+1) = \mathbf{p}(n) + \frac{\alpha_\mathrm{p}}{\|\mathbf{X}(n)\mathbf{w}(n)\|^2 + \psi_\mathrm{p}} e(n)\mathbf{X}(n)\mathbf{w}(n) \quad (11)$$

where α_p is the control parameter and ψ_p, a small constant that prevents divisions by values close to zero. Also in (11),

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

is the error signal. Similarly to (9)-(11), the following update expression can be obtained for the coefficients of the FIR filter:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\alpha_\mathrm{w}}{\|\mathbf{X}^\mathrm{T}(n)\mathbf{p}(n)\|^2 + \psi_\mathrm{w}} e(n)\mathbf{X}^\mathrm{T}(n)\mathbf{p}(n) \quad (13)$$

with α_w and ψ_w functionally similar to α_p and ψ_p.

2.2. Strategies for Practical Implementation

In practical applications, the implementation of an NLMS adaptive Hammerstein filter is carried out considering not only (11) and (13), but also certain strategies for improving the performance, obtaining better numerical properties or even reducing the computational complexity. One among these strategies is related to the calculus of the matrix product $\mathbf{X}^\mathrm{T}(n)\mathbf{p}(n)$ in (13). Note that, since $\mathbf{p}(n)$ is not available before the nth iteration, the whole product $\mathbf{X}^\mathrm{T}(n)\mathbf{p}(n)$ can only be evaluated in such an iteration. However, if one assumes that $\mathbf{p}(n)$ varies slowly, only the first element of $\mathbf{X}^\mathrm{T}(n)\mathbf{p}(n)$ needs to be evaluated at the nth iteration and, by reusing the elements obtained in previous iterations, an approximated version of $\mathbf{X}^\mathrm{T}(n)\mathbf{p}(n)$ can be obtained with smaller computational cost. Another implementation strategy adopted in adaptive Hammerstein filters is the use of the input orthogonalization procedure described in [2], which aims to improve the convergence speed. This strategy is not considered in the present paper since it is complementary to the strategy proposed here, which means that the input orthogonalization can be used concurrently with the proposed strategy in practical applications. A third adaptive Hammerstein implementation strategy is to keep constant (not to adapt) one of the filter coefficients aiming to ensure the uniqueness of the optimum coefficient values [2]. This strategy, despite being crucial to the numerical stability of the adaptive algorithm, may have a significant negative impact on the convergence speed.

3. EFFICIENT IMPLEMENTATION OF NLMS HAMMERSTEIN FILTERS

In this section, a novel effective strategy for implementing adaptive Hammerstein filters is discussed. The aim here is to obtain satisfactory numerical properties without slowing down the convergence by keeping constant one of the filter coefficients. In this context, the non-uniqueness of the optimum coefficient values and the impact of keeping one coefficient constant are initially discussed. Subsequently, the proposed strategy is described along with the analysis of its numerical properties and computational complexity.

3.1. Non-Uniqueness of the Optimum Coefficient Values

The simultaneous adaptation of both memoryless nonlinearity and FIR filter, composing an adaptive Hammerstein filter, often presents significant numerical problems. This is due to the non-uniqueness of the optimum solution, which means that there are infinite combinations of coefficient values that lead to the minimum (optimum) value of mean-square error (MSE). Such characteristic can be verified by using the property $\mathrm{vec}[\mathbf{ADB}] = (\mathbf{B}^\mathrm{T} \otimes \mathbf{A})\mathrm{vec}(\mathbf{D})$ [9] (with $\mathbf{A}$, $\mathbf{D}$, and $\mathbf{B}$ representing generic matrixes) to rewrite (8) as

$$y(n) = \mathbf{w}^\mathrm{T}\mathbf{X}^\mathrm{T}(n)\mathbf{p} = (\mathbf{p}^\mathrm{T} \otimes \mathbf{w}^\mathrm{T})\mathrm{vec}[\mathbf{X}^\mathrm{T}(n)] \quad (14)$$

with $\otimes$ representing the Kronecker product and $\mathrm{vec}[\cdot]$, the operator that converts a matrix to a vector stacking the columns of such a matrix [9]. By defining an equivalent Hammerstein coefficient vector as

$$\mathbf{h} = (\mathbf{p} \otimes \mathbf{w}) \quad (15)$$

expression (14) can be rewritten as

$$y(n) = (\mathbf{p}^{\mathrm{T}} \otimes \mathbf{w}^{\mathrm{T}})\mathbf{x}(n) = \mathbf{h}^{\mathrm{T}}\mathbf{x}(n) \quad (16)$$

with $\mathbf{x}(n) = \mathrm{vec}[\mathbf{X}^{\mathrm{T}}(n)]$ denoting the input vector. From (16) and also considering the properties of the Kronecker product [9], one can note that a given equivalent coefficient vector can be obtained from either $\mathbf{p}$ and $\mathbf{w}$ or $c\mathbf{p}$ and $\mathbf{w}/c$, with c representing an arbitrary scalar, i.e.,

$$\mathbf{h} = (\mathbf{p} \otimes \mathbf{w}) = [(c\mathbf{p}) \otimes (\mathbf{w}/c)] = (c/c)(\mathbf{p} \otimes \mathbf{w}). \quad (17)$$

Since c can assume infinite possible values, one observes, from (16) and (17), that the same output signal, and consequently the same minimum MSE, can be obtained from infinite combinations of coefficient values. In this way, situations with small values in one coefficient vector ($\mathbf{p}$, for instance) and large values in the other ($\mathbf{w}$) tend to occur frequently, leading to loss of precision (due to small values) and/or overflow (because of large values).

3.2. Implementation with a Constant Coefficient

A strategy that is commonly used to circumvent the numerical problems of standard adaptive Hammerstein filters is to keep constant the value of one of the filter coefficients [2]. As seen in [2] and [6], the chosen coefficient is usually the first from the linear filter $\mathbf{w}$, whose value is kept equal to 1. However, aiming to obtain a lower computational burden, to keep constant the first coefficient from the nonlinearity $\mathbf{p}$ is a more attractive strategy. In doing so, the evaluation of the first element of the vector resulting from $\mathbf{X}(n)\mathbf{w}(n)$ in (11) is avoided, which directly results in a reduction of N multiplications and $N-1$ additions per iteration. A similar computational saving is not observed when the first coefficient of $\mathbf{w}$ is kept constant, since the product $\mathbf{X}^{\mathrm{T}}(n)\mathbf{p}(n)$ in (13) is usually obtained by using data from previous iterations (see Section 2.2).

The expression to update $\mathbf{p}$, with its first coefficient equal to 1 using the NLMS algorithm, is obtained by including the following constraint on the minimization problem described by (9) and (10):

$$\mathbf{c}^{\mathrm{T}}\mathbf{p}(n+1) = 1 \quad (18)$$

where $\mathbf{c}$ represents a constraint vector (in this case, the first element of $\mathbf{c}$ is equal to 1 and the remaining ones equal to 0). Thus, solving the resulting minimization problem (analogously to the solution presented in [10]), one obtains

$$\mathbf{p}(n+1) = \mathbf{P}\mathbf{p}(n) + \frac{\alpha_{\mathrm{p}} e(n)}{\|\mathbf{P}\mathbf{X}(n)\mathbf{w}(n)\|^2 + \psi_{\mathrm{p}}}\mathbf{P}\mathbf{X}(n)\mathbf{w}(n) + \mathbf{c}. \quad (19)$$

where $\mathbf{P} = \mathbf{I}_M - \mathbf{c}\mathbf{c}^{\mathrm{T}}$ and $\mathbf{I}_M$ is an $M \times M$ identity matrix.

In spite of considerably improving the numerical properties of the adaptive algorithm, the adaptive Hammerstein implementation obtained by keeping constant one of the coefficients tends to present smaller convergence speed than the conventional implementation. The use of the constraint given in (18) contributes significantly to such performance degradation, since it restrains the set of possible solutions in a way that a deviation is produced with respect to the optimum convergence direction estimated by the standard adaptive algorithm at each iteration. Thus, the algorithm tends to slow down due to the accumulation of the effect of such a deviation during hundreds or thousands of iterations. Another problem that may arise from keeping constant one of the coefficients is associated with the relationship between the value of such coefficient and the values of the others. For instance, consider that the first coefficient of $\mathbf{w}$ is kept equal to 1 in a case where such coefficient is that of smaller magnitude after convergence. In this case, considering also the initialization of the other coefficients of $\mathbf{w}$ with zeros (which is a common practice), a considerable number of iterations will be required to lead these coefficients to their steady-state values, which are larger or much larger than 1. Thus, one notes that the choice of the coefficient to be kept constant may have a significant impact on the convergence speed of the adaptive algorithm.

3.3. Proposed Implementation Strategy

Aiming to obtain an adaptive Hammerstein filter with satisfactory numerical properties (without slowing down the adaptive algorithm), a novel implementation strategy is presented in this section. Such a strategy is based on keeping the Euclidean norm of one of the coefficient vectors ($\mathbf{p}$ or $\mathbf{w}$) equal to 1 throughout the adaptive process, without modifying the equivalent coefficient vector $\mathbf{h}(n+1)$ obtained at the end of each iteration. We choose to keep the norm of $\mathbf{p}(n)$ equal to 1 since this vector usually presents a smaller number of coefficients than $\mathbf{w}(n)$ in practical applications, which leads to a smaller computational burden. The proposed strategy is then implemented multiplying, at the end of each iteration, $\mathbf{p}(n+1)$ by $k = 1/\|\mathbf{p}(n+1)\|$ and $\mathbf{w}(n+1)$ by $1/k$ [with $\|\mathbf{p}(n+1)\|$ representing the Euclidean norm of $\mathbf{p}(n+1)$]. Such a procedure results in a vector $\mathbf{p}(n+1)$ with unit norm, maintaining the equivalent coefficient vector $\mathbf{h}(n+1)$ unchanged, which can be verified from the following expression:

$$\begin{aligned}\mathbf{h}(n+1) &= [k \cdot \mathbf{p}(n+1)] \otimes [(1/k) \cdot \mathbf{w}(n+1)] \\ &= (k/k) \cdot [\mathbf{p}(n+1) \otimes \mathbf{w}(n+1)] = [\mathbf{p}(n+1) \otimes \mathbf{w}(n+1)].\end{aligned} \quad (20)$$

3.3.1. Numerical Characteristics

To assess the characteristics of the solution obtained by using the proposed normalization strategy, we first consider that, if the solution is unique, the equality

$$\mathbf{h} = \mathbf{p}_{\mathrm{an}} \otimes \mathbf{w}_{\mathrm{a}} = \mathbf{p}_{\mathrm{bn}} \otimes \mathbf{w}_{\mathrm{b}} \quad (21)$$

must only hold for $\mathbf{w}_{\mathrm{a}} = \mathbf{w}_{\mathrm{b}}$ and $\mathbf{p}_{\mathrm{an}} = \mathbf{p}_{\mathrm{bn}}$, where $\mathbf{w}_{\mathrm{a}}$ and $\mathbf{w}_{\mathrm{b}}$ are instances of (5) with arbitrary values, whereas $\mathbf{p}_{\mathrm{an}}$ and $\mathbf{p}_{\mathrm{bn}}$ represent instances of (3) with arbitrary values and unit norm, i.e.,

$$\|\mathbf{p}_{\mathrm{an}}\| = \|\mathbf{p}_{\mathrm{bn}}\| = 1. \quad (22)$$

Then, from (21), we can write

$$\begin{aligned}\|\mathbf{h}\|^2 &= (\mathbf{p}_{\mathrm{an}}^{\mathrm{T}} \otimes \mathbf{w}_{\mathrm{a}}^{\mathrm{T}})(\mathbf{p}_{\mathrm{an}} \otimes \mathbf{w}_{\mathrm{a}}) = (\mathbf{p}_{\mathrm{bn}}^{\mathrm{T}} \otimes \mathbf{w}_{\mathrm{b}}^{\mathrm{T}})(\mathbf{p}_{\mathrm{bn}} \otimes \mathbf{w}_{\mathrm{b}}) \\ &= (\mathbf{p}_{\mathrm{an}}^{\mathrm{T}} \otimes \mathbf{w}_{\mathrm{a}}^{\mathrm{T}})(\mathbf{p}_{\mathrm{bn}} \otimes \mathbf{w}_{\mathrm{b}})\end{aligned} \quad (23)$$

which, considering the mixed-product rule of the Kronecker product [9], results in

$$\mathbf{p}_{\mathrm{an}}^{\mathrm{T}}\mathbf{p}_{\mathrm{an}} \otimes \mathbf{w}_{\mathrm{a}}^{\mathrm{T}}\mathbf{w}_{\mathrm{a}} = \mathbf{p}_{\mathrm{bn}}^{\mathrm{T}}\mathbf{p}_{\mathrm{bn}} \otimes \mathbf{w}_{\mathrm{b}}^{\mathrm{T}}\mathbf{w}_{\mathrm{b}} = \mathbf{p}_{\mathrm{an}}^{\mathrm{T}}\mathbf{p}_{\mathrm{bn}} \otimes \mathbf{w}_{\mathrm{a}}^{\mathrm{T}}\mathbf{w}_{\mathrm{b}}. \quad (24)$$

Taking into account that the Kronecker product between two scalars is equal to a multiplication, squaring (24), and taking the square root of the resulting expression, one obtains

$$\| \mathbf{p}_{\mathrm{an}} \|^2 \cdot \| \mathbf{w}_{\mathrm{a}} \|^2 = \| \mathbf{p}_{\mathrm{bn}} \|^2 \cdot \| \mathbf{w}_{\mathrm{b}} \|^2 = | \mathbf{p}_{\mathrm{an}}^{\mathrm{T}} \mathbf{p}_{\mathrm{bn}} | \cdot | \mathbf{w}_{\mathrm{a}}^{\mathrm{T}} \mathbf{w}_{\mathrm{b}} |. \quad (25)$$

From (22) and (25), one has

$$\| \mathbf{w}_{\mathrm{a}} \|^2 = \| \mathbf{w}_{\mathrm{b}} \|^2 = | \mathbf{p}_{\mathrm{an}}^{\mathrm{T}} \mathbf{p}_{\mathrm{bn}} | \cdot | \mathbf{w}_{\mathrm{a}}^{\mathrm{T}} \mathbf{w}_{\mathrm{b}} |. \quad (26)$$

By considering now the Cauchy-Schwarz inequality [11], and also (22) and (26), we can write

$$| \mathbf{p}_{\mathrm{an}}^{\mathrm{T}} \mathbf{p}_{\mathrm{bn}} | \le \| \mathbf{p}_{\mathrm{an}} \| \cdot \| \mathbf{p}_{\mathrm{bn}} \| = 1 \quad (27)$$

and

$$| \mathbf{w}_{\mathrm{a}}^{\mathrm{T}} \mathbf{w}_{\mathrm{b}} | \le \| \mathbf{w}_{\mathrm{a}} \| \cdot \| \mathbf{w}_{\mathrm{b}} \| = \| \mathbf{w}_{\mathrm{a}} \|^2. \quad (28)$$

Note that, if $| \mathbf{p}_{\mathrm{an}}^{\mathrm{T}} \mathbf{p}_{\mathrm{bn}} | \ne 1$ in (27), $| \mathbf{p}_{\mathrm{an}}^{\mathrm{T}} \mathbf{p}_{\mathrm{bn}} | < 1$ and, in this case, (26) only holds if

$$| \mathbf{w}_{\mathrm{a}}^{\mathrm{T}} \mathbf{w}_{\mathrm{b}} | > \| \mathbf{w}_{\mathrm{a}} \|^2. \quad (29)$$

However, one observes that (29) contradicts (28) and, thus, one concludes that (27) necessarily results in

$$| \mathbf{p}_{\mathrm{an}}^{\mathrm{T}} \mathbf{p}_{\mathrm{bn}} | = 1. \quad (30)$$

Furthermore, considering (26) and (30), one observes that (28) results in

$$| \mathbf{w}_{\mathrm{a}}^{\mathrm{T}} \mathbf{w}_{\mathrm{b}} | = \| \mathbf{w}_{\mathrm{a}} \|^2. \quad (31)$$

From (30), one notices that equality holds in (27) (a Cauchy-Schwarz inequality) and, consequently, the involved vectors ($\mathbf{p}_{\mathrm{an}}$ and $\mathbf{p}_{\mathrm{bn}}$) are linearly dependent [11]. Thus, we can write

$$\mathbf{p}_{\mathrm{an}} = \beta \mathbf{p}_{\mathrm{bn}} \quad (32)$$

and therefore

$$\| \mathbf{p}_{\mathrm{an}} \| = | \beta | \cdot \| \mathbf{p}_{\mathrm{bn}} \| \quad (33)$$

with β representing an arbitrary scalar. By considering now (22), one notices that (33) only holds for $\beta = \pm 1$, implying that

$$\mathbf{p}_{\mathrm{an}} = \pm \mathbf{p}_{\mathrm{bn}}. \quad (34)$$

Similarly, one can conclude that

$$\mathbf{w}_{\mathrm{a}} = \pm \mathbf{w}_{\mathrm{b}}. \quad (35)$$

From (21), (34), and (35), one observes that the proposed strategy, despite not leading to a unique solution for $\mathbf{p}$ and $\mathbf{w}$, provides two possible solutions differing only by the signs of the coefficients and not by their magnitude. Thus, the risk of having very small values of coefficients in one of the coefficient vectors and very large values in the other is considerably reduced, implying satisfactory numerical properties for the algorithms based on the proposed strategy.

3.3.2. Computational Complexity

The use of the proposed strategy results in an increase of $2M + N$ multiplications, $M - 1$ additions, 1 division, and 1 square-root operation with respect to the conventional Hammerstein filter implementation. Such an increase is relatively small as can be seen in Fig. 2. In this figure, curves of the number of operations per sample as a function of the nonlinearity order are shown for the following implementations of NLMS Hammerstein filters with memory size $N = 100$ (considering the cost of 1 division or 1 square root equal to that of 10 multiplications or 10 additions): (I) conventional implementation; (II) with the first coefficient of the nonlinearity kept equal to 1; (III) with the strategy of normalizing the coefficient vectors (proposed strategy); and (IV) with the normalization of the coefficient vectors performed only at each 10 iterations [reduced-complexity version of (III)]. Implementation (IV) is obtained by adapting the filter coefficients for 9 iterations and performing the normalization on the tenth iteration. Thus, the curve corresponding to (IV) in Fig. 2 represents the average number of operations per sample. Besides the curves for the four implementations of Hammerstein filters, Fig. 2 also shows a curve of the number of operations per sample [indicated by (V)] for the power filter using a non-adaptive version of the input-orthogonalization procedure described in [12]. Such a filter is considered here for comparison purposes since the Hammerstein filter can be seen as a particular case of the power filter [12].

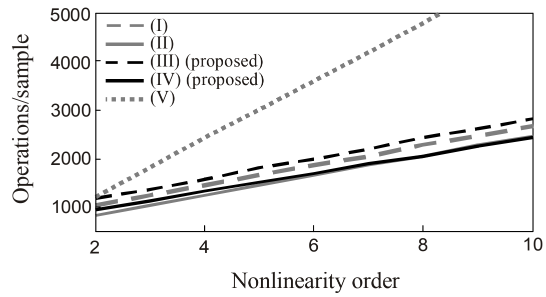

Fig. 2. Computational complexity required by different implementations of NLMS Hammerstein filters and of the NLMS power filter.

4. SIMULATION RESULTS

In this section, numerical simulation results are presented aiming to compare the performance of two adaptive implementations of the Hammerstein filter using the proposed strategy [(III) and (IV)] with those of the other implementations described in Section 3.3.2 [Hammerstein implementations indicated by (I) and (II) as well as the power filter indicated by (V)]. Such a comparison is performed in terms of MSE curves obtained from Monte Carlo simulations (average of 100 independent runs) involving the modeling of Hammerstein plants. The linear part of the plants considered here presents the impulse response shown in Fig. 3 (a scaled version of the m_2 response from the ITU G.168 recommendation [13]). The adaptive filters used to model the plant present a memory size of 100. The input signal used is white Gaussian with unit variance and the measurement noise, added to the output of the plant, has variance $\sigma_z^2 = 10^{-6}$.

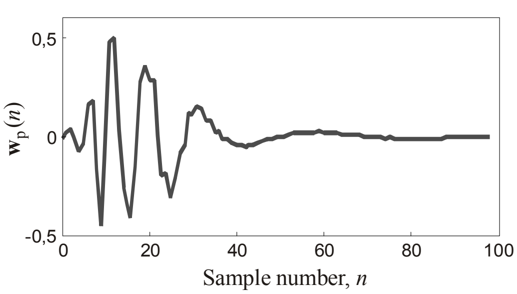

Fig. 3. Impulse response of the linear part of the plant.

4.1. Example 1

The nonlinearity of the plant considered in this example is described by $\hat{x}(n) = 0.5x(n) - 0.15x^2(n) + 0.05x^3(n)$. The parameters of the NLMS algorithm used for (I)-(IV) are: $\alpha_g = \alpha_h = 0.25$, $\psi_h = 1$, and $\psi_g = 10$. For (V), the parameters of the NLMS algorithm are $\alpha = 0.5$ and $\psi = 10^{-3}$. The obtained MSE curves are shown in Fig. 4. One observes the very good performance of the implementations based on the proposed strategy [(III) and (IV)]. These implementations provide practically the same convergence speed than (I) (conventional implementation presenting numerical problems). Moreover, such convergence speed is considerably superior than that obtained by using (II) (implementation with one coefficient kept equal to 1) and (V) (power filter). In addition, it is important to highlight that the difference of performance between (III) and its reduced-complexity version (IV) is very small for this example.

4.2. Example 2

The plant used in this example has a nonlinearity described by $\hat{x}(n) = 0.1x(n) + 0.2x^2(n)$. The parameters of the NLMS algorithm used by (I)-(IV) are: $\alpha_h = 0.3$, $\alpha_g = 0.1$, $\psi_h = 1$, and $\psi_g = 10$. For (V), the parameters of the NLMS algorithm are the same as in Example 1. The obtained MSE curves are shown in Fig. 5. We again notice the very good performance of the implementations based on the proposed strategy [(III) and (IV)] especially in comparison with the poor performance of (II) (implementation with one coefficient kept equal to 1). These results attest the efficiency of the implementations based on the proposed strategy and, in particular, the attractiveness of (IV) due to its reduced complexity in comparison with (III).

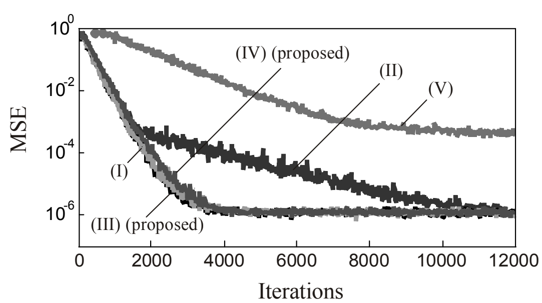

Fig. 4. MSE curves for Example 1.

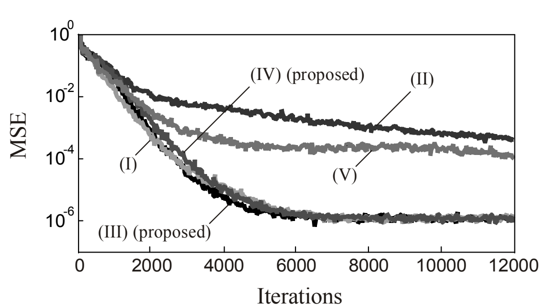

Fig. 5. MSE curves for Example 2.

5. CONCLUDING REMARKS

In this paper, a novel strategy to implement adaptive Hammerstein filters was discussed. Such a strategy is based on using a procedure to normalize the coefficient values aiming to avoid numerical problems that arise from adapting the cascaded structure of a Hammerstein filter. As a result, new adaptive Hammerstein implementations were obtained, leading to a superior performance as compared with the implementation carried out by keeping constant one of the coefficients. Numerical simulation results were shown corroborating the effectiveness of the proposed strategy.

6. REFERENCES

[1] V. J. Mathews and G. L. Sicuranza, *Polynomial Signal Processing*, John Wiley & Sons Inc., 2000.

[2] J. Jeraj and V. J. Matthews, "A stable adaptive Hammerstein filter employing partial orthogonalization of the input signals," *IEEE Trans. Signal Process.*, vol. 54, no. 4, pp. 1412-1420, Apr. 2006.

[3] J.-P. Costa, A. Lagrange, and A. Arliaud, "Acoustic echo cancellation using nonlinear cascade filters," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Hong Kong, China, vol. 5, Apr. 2003, no. 1, pp. V-389-92.

[4] J. Fu and W.-P. Zhu, "A nonlinear acoustic echo canceller using sigmoid transform in conjunction with RLS algorithm," *IEEE Trans. Circ. Syst. II: Express Briefs*, vol. 55, no. 10, pp. 1056-1060, Oct. 2008.

[5] K. Shi, X. Ma, and G. Tongzhou, "An efficient acoustic echo cancellation design for systems with long room impulses and nonlinear loudspeakers," *Signal Processing*, vol. 89, no. 2, pp. 121-132, Feb. 2009.

[6] J. Jeraj and V.J. Mathews, "Stochastic mean-square performance analysis of an adaptive Hammerstein filter," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2168-2177, Jun. 2006.

[7] B. Farhang-Boroujeny, *Adaptive Filters Theory and Applications*. Wiley, 1999.

[8] 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. Eur. Signal Process. Conf. (EUSIPCO)*, Poznan, Poland, Sept. 2007, pp. 370-374.

[9] J. W. Brewer, "Kronecker Products and Matrix Calculus in System Theory," *IEEE Trans. Circ. Syst.*, vol. CAS-25, no. 9, pp. 772-781, Sept. 1978.

[10] E. L. O. Batista and Rui Seara, "On The Implementation of Fully Adaptive Interpolated FIR Filters," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Bucharest, Romania, Aug. 2012, pp. 2173-2177.

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

[12] F. Kuech, A. Mitnacht, and W. Kellermann, "Nonlinear acoustic echo cancellation using adaptive orthogonalized power filters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.* (*ICASSP*), Philadelphia, PA, Mar. 2005, vol. 3, pp. 105-108.

[13] ITU-T Recommendation G.168, *Digital Network Echo Cancellers*, June 2002.