SIGNAL PROCESSING VII: Theories and Applications
M. Holt, C. Cowan, P. Grant, W. Sandham (Eds.)

The K-filter: Design alternatives to model non-linear systems

Alba PAGÈS-ZAMORA, Miguel A. LAGUNAS

Dept. Teoria del Senyal i Comunicacions, Universitat Politècnica de Catalunya., Barcelona, SPAIN, Tel/Fax: 93 - 401 64 54 / 401 64 47, E-mail: alba@tsc.upc.es

Abstract. This paper presents an architecture named K-filter able to model non-linear systems both memoryless and with memory. The most general version of the k-filter applies to any non linearity but sometimes at the cost of holding a considerable computational load, specially when the memory of the non-linear system increases. Thus, the paper is basically devoted to present how different simpler versions of the original k-filter can be obtained taking into account symmetrical properties of the input/output relation of the non-linear system to model. The theoretical points along with the simulation results will show how these symmetrical considerations simplify the k-filter without making worse the performance.

1. Introduction*

Recently, the presence of non-linear functions in signal processing has increased surprisingly; such outstanding subjects as neural networks, chaotic series and high order statistics among others are strongly related to non-linear processing field. On the other hand, the architectures used to model non-linear systems (NLSs) are basically the ones proposed by Schetzen, based on the Volterra and Wiener functionals [1]. This paper presents the k-filter as a new option to model NLSs [2]. An important feature is the simplicity of the theoretical development done to obtain the k-filter, specially the basic mathematical tools used. The consequence is a direct relation between the parameters of the design and the characteristics of the real system. This fact provides the k-filter with the capability of being easily modified in front of different systems.

The proposed architecture estimates the output of a given NLS by a linear combination of a family of non-linear functions, which consists in complex exponential transformations of the input signal. From this point of view, a similar structure can be found in Mulgrew's work [3] that relates the k-filter to the estimation of the output of an NLS in terms of orthonormal functions [1]. The theoretical approach followed in this paper to obtain the k-filter is completely different from the way in which Mulgrew focuses the problem. From the point of view of the authors, both works are complementary in the sense that they reinforce the final architecture.

The third section includes the main topic of this work: the so-called *odd k-filter*, that consists in a different version of the original k-filter [4,5]. Nevertheless, it has been considered interesting to start with a review of the k-filter before arriving to the odd k-filter in order to give an overall view of the model.

2. The k-filter

This second section is devoted to present different schemes of the k-filter beginning by the memoryless one to the case with memory. All of them come from the same approach but particularized to different situations.

2.1. The memoryless k-filter

The modelling of a memoryless NLS is the most elemental case found when dealing with NLSs but, at the same time, is the situation that gives more intuitive information about the model used, i.e. structure, parameters of design, gains/drawbacks...

* This work has been partially supported by a grant of the Generalitat de Catalunya and by Grant TIC-92-0800-C05-05.

Assume g[.] denotes the input/output (in/out) relation of an NLS. In case of dealing with a memoryless one, the function is one-dimensional and the output is equal to y(t)=g[x(t)], being x(t) the input signal. Then, the output can be approximated by developing g[.] in terms of a Fourier series, eq.(1,2).

$$\hat{y}(t) = \hat{g}[x(t)] = \sum_{n=0}^{N} G_n \cdot \exp[jn\omega_0 x(t)] \quad (1)$$

$$G_n = F[g(x)]\Big|_{\omega=n\omega_0} \qquad \omega_0 \le \frac{\pi}{X\max} \quad (2)$$

In fact, the approximation is a linear combination of exponential transformations of the input but the coefficients of this linear combination, G_n, are determined by the NLS. If eq.(1) represents a Fourier series, the coefficients must be related to the Fourier transform of the function that is developed. Eq.(2) shows how they coincide with the values of sampling the Fourier transform of g[x] at multiples of the principal frequency, ωo.

The approximation $\hat{y}(t)$ can be viewed as the output of our model that is a new memoryless NLS characterized by the in/out relation $\hat{g}[.]$. This function is periodic in the x domain, being the period repetition equal to To=(2π/ωo). Then, assuming the input signal belongs to the range [-Xmax, Xmax], the frequency of the Fourier series must be bounded to ωo<=(π/Xmax) in order to avoid aliasing. In spite of being $\hat{g}[.]$ and g[.] completely different out of the range [-Xmax, Xmax], the output $\hat{y}(t)$ is a "suitable" approximation of the output of the real NLS, y(t), whenever the input signal remains in the previous range.

The structure that implements the approximation denoted by eq.(1) is called the memoryless k-filter (Figure 1). It consists in applying an exponential transformation to the input signal and then to compute a polynomial series, which can be viewed as a memoryless Volterra series.

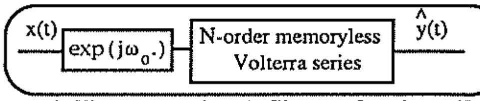

Figure 1. The memoryless k-filter performing a Fourier series of N terms.

It is important to remark that in a modelling problem where the in/out relation g[.] is known, the coefficients of the Fourier series can be computed with the Fourier transform of this function, eq.(2). On the other hand, the parameters to be determined in an identification problem are also these coefficients. Note that, as the output of the k-filter is a linear combination of them, a mean square error criterion will lead the system to the Wiener solution. Furthermore, all the adaptive techniques based in gradient or recursive methods can be also used to design the k-filter in an on-line problem. This important property of the k-filter is also shared by the k-filter with memory and all the versions included in this paper.

2.2. *The k-filter with memory.*

When facing to the problem of modelling NLSs with memory, the first idea is to generalize the memoryless k-filter (Figure 1) to a new scheme that includes memory. Thus, the k-filter with memory consists in applying the exponential transformation of the memoryless k-filter not only to the input x(t) but also to previous values of it. As it can be inferred from Figure 2, the memory in the k-filter is supplied by a temporal diversity vector of the input signal, [x(t), x(t-τ),..., x(t-(Q-1)τ].

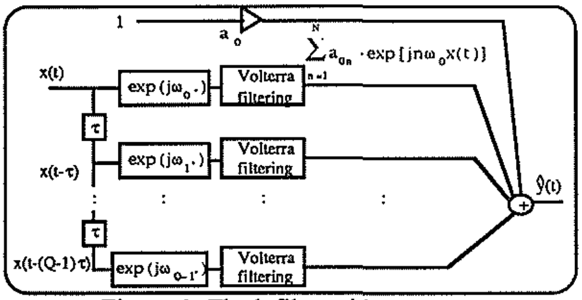

Figure 2. The k-filter with memory.

The approximation performed by the k-filter with memory can be expressed as follows:

$$\hat{y}(t) = a_0 + \sum_{q=0}^{Q-1}\left(\sum_{n=1}^{N} a_{qn}\cdot\exp(jn\omega_q x(t-q\tau))\right) \tag{3}$$

Nevertheless, this intuitive result must be supported by a quantitative approach similar to the one developed for the memoryless case. Following the same steps as in subsection (2.1), when the NLS to model has memory, the in/out relation results in a Q-dimensional (Q-dim) function. That is, the output depends not only on the actual value of the input signal, but also on (Q-1) previous values, eq.(4).

$$y(t) = g[x(t), x(t-\tau), \ldots, x(t-(Q-1)\tau)] \tag{4}$$

The Q parameter is the memory of the NLS: the larger Q is, the longer the memory of the NLS is. Note that the formulation of the problem is similar to the one of the memoryless case except that now it is a Q-dim instead of a one-dim problem.

Given eq.(4), a possible approach of the output would consist in developing g[.] in terms of a Q-dim Fourier series as it is expressed in eq.(5). Similarly to the memoryless case, the coefficients of this series, eq.(6), are equal to the values obtained from sampling the Q-dim Fourier transform of g[.] at multiples of the principal frequencies, ωi.

$$\hat{y}(t) = G_0 + \sum_{n_0=1}^{N_0}\sum_{n_1=1}^{N_1}\cdots\sum_{n_{(Q-1)}}^{N_{(Q-1)}}\left\{ G_{n_0,n_1,\ldots,n_{(Q-1)}}\cdot e^{-j\left[n_0\cdot\omega_0\cdot x(t)+n_1\cdot\omega_1\cdot x(t-\tau)+\ldots+n_{(Q-1)}\cdot\omega_{(Q-1)}\cdot x(t-(Q-1)\tau)\right]}\right\} \tag{5}$$

$$G_{n_0,n_1,\ldots,n_{(Q-1)}} = F\{g[x(t),\ldots,x(t-(Q-1)\tau)]\}\Big|_{\{n_0\cdot\omega_0,n_1\cdot\omega_1,\ldots,n_{(Q-1)}\cdot\omega_{(Q-1)}\}}$$

$$n_i = 1\ldots Ni \quad \omega_i \geq \frac{\pi}{X\max} \qquad \forall i = 0\ldots(Q-1) \tag{6}$$

At first glance the output of the k-filter, eq.(3), and the approximation obtained from the mathematical approach, eq.(5), seem to be completely different. But actually, the k-filter is related to the more general development of eq.(5) because the coefficients aqn of eq.(3) are samples of the axis of the Q-dim Fourier transform of g[.], being equal to certain Gn as it is denoted by eq.(7).

$$a_{qn} = G_{0,\ldots,0,n_q,0,\ldots,0} \qquad \forall n = 1..N, q = 0..(Q-1) \tag{7}$$

In consequence, it is possible to view the k-filter (Figure 2, eq.(3)) as a truncated version of the Q-dim Fourier series of eq.(4). The possibility of designing a modified k-filter that implements eq.(5) instead of eq.(3) is discussed in [5]. The structure is based on the k-filter of Figure 2 but inserting a functional module that combines exponentials of different temporal components of the input vector. The result is a new k-filter named *k-filter with modulation combiner* which obviously requires much more computational load than the other one, specially if Q increases. But, on the other hand, the k-filter with modulation combiner include cross terms between different temporal components of the input vector whereas the k-filter does not. This fact is obviously a drawback of the k-filter compared with another models that include these cross terms as the Volterra functionals. In this sense, the k-filter with modulation combiner is necessary to provide the k-filter with a complete structure, able to face a wide variety of problems although at the cost of a high computational load.

2.3. *The aperiodic k-filter*

From the previous subsections, it can be inferred that the k-filter implements an NLS with an in/out function denoted by $\hat{g}[.]$ which is onedimensional if it models a memoryless NLS and multidimensional in the case with memory. The function $\hat{g}[.]$ is periodic in the input domain and this property can be not desirable in situations such as the analogical amplifiers that the range of the input signal is not perfectly known. Hence, the purpose of this point is to find a filter that implements an aperiodic non-linear function g'[.] that approximates the actual non-linear function g[.] in the whole axis of x and not only in the range of [-Xmax, Xmax]. The process will be developed in the memoryless case, which considers g[x] as a one-dim function, but the generalization of the final structure to the Q-dim case when the NLS has memory can be found in [2,4,5].

Assume G(ω) is the Fourier transform of g[x], the in/out function of a memoryless NLS. Then, it is possible to build a function g'[x] that consists in the inverse Fourier transform of sampling G(ω) randomly as eq.(8) denotes.

$$y'(t) = g'[x(t)] = \int_{-\infty}^{+\infty}\left[\sum_{n=-\infty}^{+\infty} G(f)\cdot\delta(f-f_n)\right]\cdot\exp(j2\pi fx)\,df \tag{8}$$

Bilinskis shows how using an appropriate random sampling to sample the Fourier transform , the expected value of g'[x] is equal to the real function g[x], [6]. This result implies that although the Fourier transform is sampled, the mean value of the function in the x domain does not suffer from aliasing o repetition.

$$E[y'(t)] = \int_{-\infty}^{+\infty} G(f)\, E\left[\sum_{n=-\infty}^{+\infty}\delta(f-f_n)\right]\cdot\exp(j2\pi fx)\,df$$

$$E\left[\sum_{n=-\infty}^{+\infty}\delta(f-f_n)\right] = \text{constant} \quad\Rightarrow\quad E[y'(t)] = y(t) \tag{9}$$

The difficulty in this problem is to find the architecture that will perform the random sampling of eq.(8). An important characteristic of the sample frequencies fn is pointed out by Bilinskis. This variable is not itself a random variable, but the difference between two consecutive samples must be the random variable. Taking into account this fact, the following modified memoryless k-filter of Figure 3 is proposed to model the inverse of a function that samples randomly the Fourier transform of g[x].

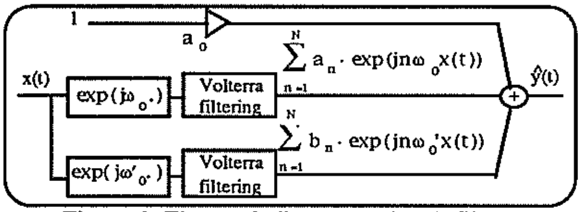

Figure 3. The aperiodic memoryless k-filter.

The output of the aperiodic k-filter consists in two N-order Fourier series computed using two different principal frequencies, ωo and ωo'. These frequencies must be chosen in order to simulate the random sampling in the transformed domain [2].

3. Different approaches of the k-filter considering symmetrical properties

Each one of the schemes presented in section 2 implements a truncated version of the development in terms of a Fourier series of the in/out relation of an NLS, g[.]. Hence, symmetrical properties of g[.] could be used to simplify the Fourier series and, in consequence, the k-filter.

3.1. Basic symmetrical properties of g[.]

In case of dealing with a memoryless NLS that has an odd function g[.], the approximation of this function in terms of a Fourier series can be reduced to a sinus series, eq.(10).

$$g[x(t)] = -g[-x(t)] \quad \forall x(t) \;\Rightarrow\; \hat{y}(t) = \sum_{n=1}^{N} a_n \cdot \sin[n\omega_0 x(t)] \qquad (10)$$

The memoryless k-filter of Figure.1 can be then modified to implement the sinus development instead of the Fourier series with exponentials of eq.(1). Whereas the performance holds because both are equivalent, the computational load is clearly reduced due to the real nature of eq.(10), both the coefficients and the sinus functions.

This property also applies to the k-filter with memory where the condition of odd symmetry of the Q-dim in/out relation of the NLS to model is denoted by eq.(10). As previously, the approximation is built by reducing the Q-dim Fourier series of eq.(5) in the Q-dim sinus series of eq.(11).

$$g[x(t),\ldots,x(t-(Q-1)\tau)] = -g[-x(t),\ldots,-x(t-(Q-1)\tau)] \qquad (10)$$

$$\hat{y}(t) = a_0 + \sum_{n_0=1}^{N_0} \sum_{n_1=1}^{N_1} \cdots \sum_{n_{(Q-1)}=1}^{N_{(Q-1)}} \Big\{ a_{n_0,n_1,\ldots,n_{(Q-1)}} \cdot \sin\big(n_0\omega_0 x(t) + n_1\omega_1 x(t-\tau) + \ldots + n_{(Q-1)}\omega_{(Q-1)} x(t-(Q-1)\tau)\big) \Big\} \qquad (11)$$

As it was done in subsection (2.2), the k-filter does not include all the coefficients of the Q-dim Fourier series, only the ones that correspond to values taken from the axis of the Fourier transform. Thus, the approximation that the k-filter performs when g[.] has odd symmetry is reduced to eq.(12).

$$\hat{y}(t) = a_0 + \sum_{q=0}^{Q-1} \left(\sum_{n=1}^{N} a_{qn} \cdot \sin(n\omega_q x(t-q\tau)) \right) \qquad (12)$$

Figure 4 shows the k-filter with memory when the condition of eq.(10) applies. It could also implement the complete approximation of eq.(11) instead of the simplified version of eq.(12) just by inserting a functional module before the sinus transformations that combines different temporal components of the input vector. The resulting architecture would result in the k-filter with modulation combiner when g[.] is an odd Q-dim function.

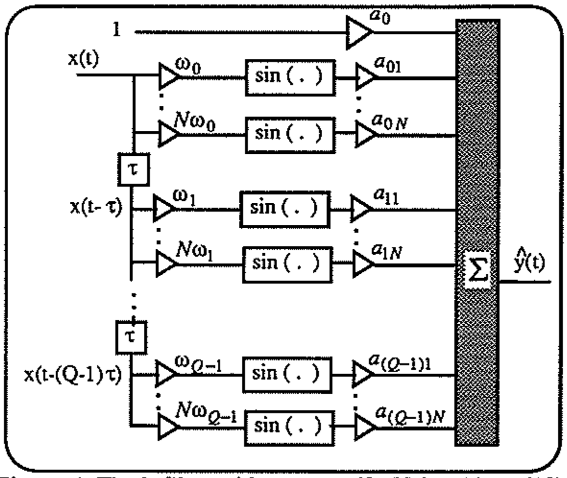

Figure 4. The k-filter with memory if g[.] is odd. eq.(10)

It is worthwhile to point out the case of having an in/out function g[.] with even symmetry. The development would be analogue to the one done for the odd case except that cosines functions would be obtained instead of sinus ones.

3.2. The odd k-filter

A part from the odd and even symmetries of function g[.], the authors have worked on the possibility of building an "artificial" function g'[.] that is equal to the real one in the range of [-Xmax,Xmax] but that also has symmetric properties which simplify even more the k-filter. This function has been obtained taking into account different values of the sampling frequency. As it is well known, when the Fourier transform of certain function is sampled, this function is repeated with a period inversely proportional to the sampling period used in the transformed domain. If the repetition period is longer than the length of the function, the empty sections are padded with zeros. Clearly the Gibbs phenomenon is a drawback in the k-filter, furthermore when the NLS to model shows a saturation effect. The question that the authors asked themselves was how this empty section of the rebuilt signal can be padded in order to simplify the Fourier series. The proposed function g'[x] is formulated in eq.(13) in function of the real in/out relation of the NLS g[x], which is bounded to the range [-Xmax, Xmax]. Thus, the resulting Fourier transform G'(ω) is related to G(ω) at multiples of the sampling frequency ωo as it denotes eq.(14).

$$g'(x) = \begin{cases} -g(x+2X\max) & -2X\max < x < -X\max \\ g(x) & -X\max < x < X\max \\ -g(x-2X\max) & X\max < x < 2X\max \end{cases} \qquad (13)$$

$$G'(n\omega_0) = \begin{cases} 0 & n \text{ even} \\ 2G(n\omega_0) & n \text{ odd} \end{cases} \qquad \omega_0 = \frac{\pi}{2X\max} \qquad (14)$$

Note that the output of an NLS with an in/out function g'[.] and another with g[.] would be the same whenever x(t) belongs to the range [-Xmax, Xmax] but the Fourier series development of g'[.] only needs the odd harmonics. The resulting scheme of the memoryless k-filter would be equal to Figure 1 except that the Volterra filtering only would include odd terms. In case of dealing with an NLS with memory where g[.] is Q-dim, eq.(13,14) can be easily generalized to the Q-dim problem. As it has been inferred in the memoryless case, the resulting k-filter with memory would be equal to Figure 2 except that the Volterra filtering only would include odd terms. The resulting approximation is denoted by eq.(15), which is the output of the so-called *odd k-filter*.

$$\hat{y}(t) = a_0 + \sum_{q=0}^{Q-1}\left(\sum_{n=1}^{N} a_{q(2n+1)} \cdot \exp(j(2n+1)\omega_q x(t-q\tau))\right) \quad (15)$$

It is possible to verify that the development of g'[.] in terms of a Fourier series also allows a simplification to a sinus series when it is an odd function. According to eq.(13), g'[x] is odd when g[x] is also odd. Thus, in the case with memory, eq.(15) is simplified to a Q-dim sinus series with only the odd harmonics. Any sinus of an odd multiple of a certain frequency can be expressed as an addition of odd powers of the sinus of this frequency. Then, the k-filter with memory allows the following structure to implement the Q-dim sinus series of odd harmonics.

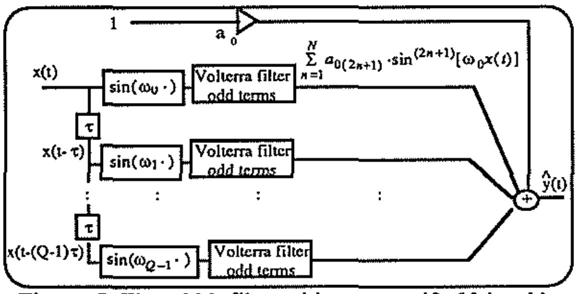

Figure 5. The odd k-filter with memory if g[.] is odd.

4. Simulation results.

This section is devoted to basically compare the k-filter with memory eq.(3) with the odd-filter, with exponentials eq.(13) and with sinus (Figure 5). The simulations consist in identifying (adaptatively or not) a communication channel where the amplifiers of the transmitter and receiver show non-linear effects. The input has been chosen as a sampled band-pass normal distributed noise filtered through the system, $H(z)=(z^{-4}+2.7607z^{-3}+3.8106z^{-2}+2.6535z^{-1}+0.9238)^{-1}$. The channel is modelled by the filter $Hch(z)=(z^{-3}+0.0928z^{-2}-0.3158z^{-1}+0.2)/(z^{-1}-0.5)$ and an additive gaussian noise of 10 dB has been also taken into account just at the output of the channel, before the amplifier of the receiver.

Two different functions have been considered to model the amplifiers of the system in order to test the behaviour of the odd k-filter. The first function is odd, $g[x]=\text{sign}(x)\exp(x^2/0.01)$, whereas the second one has not symmetry and consist in a polynomial relation, $g(x)=0.3x^4+0.8x^2+x$ for the amplifier of the transmitter and $g(x)=0.7x^2+x$ for the receiver. It is important to remark that although the amplifiers are memoryless, the channel includes memory to the whole system. Hence, we face with the problem of identifying an NLS with memory.

The simulations of Figure 6 show the evolution of the mean square error (MSE) of the approximation when the communication system is identified adaptatively. The weights of the filter that identifies the system are updated by the NLMS algorithm with an step-size parameter denoted by μ. Figure 6a is to the simulation done when the amplifiers are modelled with the odd function, whereas Figure 6b corresponds to the polynomial model. Five different filters have been checked to compare the results. First of all, the k-filter(Q=2,N=8) with $\omega o=\pi/X\max$ and 49 coefficients is included. Then, two odd k-filters(2,16) with $\omega o=\pi/2X\max$ are also checked: one with exponentials (49 coefficients) and the other one with sinus (25 coefficients). Finally, a FIR filter of 49 coefficients is considered along with a 5th-order Volterra filter of 2 delays (56 coefficients). Figure 6 also includes the MSE got when an off-line design is used (Wiener solution).

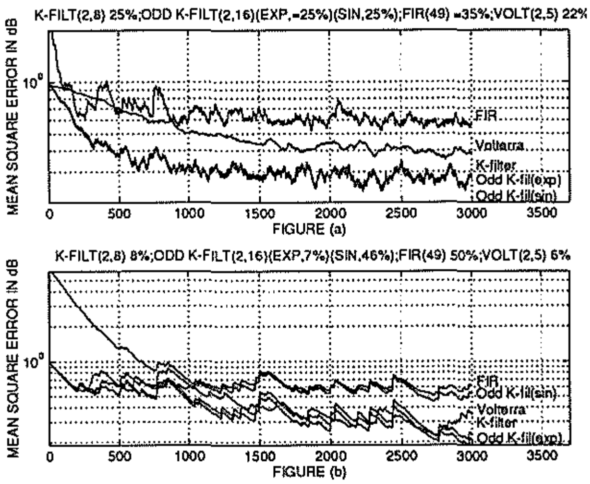

Figure 6. Evolution (3000 samples) of the MSE of a FIR(49), Volterra(2,5), k-filter(2,8) and two odd k-filters(2,16) one with exponentials and the other one with sinus. Except the Volterra filter with μ=0.01, the rest of them have μ=0.1. The % values included in the titles are the MSE values computed with an off-line solution.

Different aspects can be pointed out from the results of Figure 6. First of all, the k-filter and the odd k-filter show a better performance in the adaptive solution than the FIR filter and the Volterra filter. Secondly, the odd k-filter with sinus only has good results in Figure 6a where an odd function models the amplifier. On the other hand, the odd k-filter with exponentials has a similar performance to the k-filter in both simulations as it was suspected from the theoretical approach.

5. Remarks

This paper presents a wide overview of the so-called k-filter [2] along with different versions of it. The aim of the authors have been to give a global sense to all these architectures by showing how they come from the same approach adapted to different situations. An special attention is paid to the odd k-filter because, as it is justified theoretically and with simulation results, it shows the same performance as the k-filter but with a considerably less computational load.

References

[1] **M. Schetzen (1980):** The Volterra/Wiener theories of nonlinear systems, John Wiley & Sons.

[2] **M.A. Lagunas (1994):** The Kolmogorov mapping theorem in signal processing, *Proc. 7th SP Workshop on SSAP* (to be published), Quebec (Canada), June 1994.

[3] **B. Mulgrew (1994):** Orthonormal function for nonlinear signal processing and adaptative techniques, *Proc. Int. Conf. ICASSP-94, vol.II*, 509-512, Adelaide (Australia), April 1994.

[4] **A. Pagès-Zamora and M.A. Lagunas (1994):** A novel architecture to model nonlinear systems, *Proc. Int. Conf. ICASSP-94, vol.II*, 509-512, Adelaide (Australia), April 1994.

[5] **A. Pagès-Zamora and M.A. Lagunas (1994):** A new model of nonlinear systems with memory, *Proc. 7th SP Workshop on SSAP* (to be published), Quebec (Canada), June 1994.

[6] **I. Bilinskis and A. Mikelsons (1992):** "Randomised Signal Processing", Prentice Hall.