

THE K-FILTER: A NEW MODEL OF NON-LINEAR SYSTEMS WITH MEMORY

A. Pagès-Zamora, Miguel A. Lagunas

Grup de Processament del Senyal

Dpt de Teoria del Senyal i Comunicacions, Universitat Politècnica de Catalunya

Apt 30002, 08071 Barcelona, Spain

E-mail: alba@tsc.upc.es

ABSTRACT*

This paper deals with the problem of modelling and identifying non-linear systems (NLSs). Although the final architecture applies to NLSs with memory, it is also important to bear in mind that the proposed filter consists in a generalisation of a previous scheme which was designed to model memoryless NLSs. In consequence, the simplicity of the memoryless filter is useful to intuitively understand the final architecture and how it can be improved when dealing with specific problems. On the other hand, the filter with memory shows the complexity and computational load that non-linear theory usually involves. But at the same time, it allows a large number of different possibilities depending on the concrete application and it also opens new ways of future work.

1. INTRODUCTION

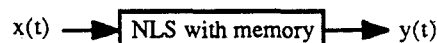
An increasing motivation centred on non-linear processing has emerged in the last ten years due, in part, to the saturation of linear processing field. At the same time some emerging new methods based on high-order statistics, useful for non-linear analysis [1], new concepts such as chaotic series, strongly related to non-linearities [2] along with another subjects have contributed to this new trend. The authors would like to include in this context the work that is presented below. It can be viewed as a contribution to the non-linear processing at the architecture level since the filter that we propose is able to model NLSs both memoryless and with memory.

As it will be seen along the paper this filter is based on the approximation of the input/output relation of the NLS to be modelled. This approximation consists in a truncated Fourier series of those input/output function, becoming a multidimensional Fourier series when the system includes memory. Furthermore, this Fourier series is obtained from a nonuniform sampling in the transformed domain. By means of this sampling process, the authors would like to emulate the result of sampling the transformed domain with a discrete variable whose difference between two consecutive values is random [3]. The resulting filter follows the main guide lines of the Kolmogorov's mapping theorem [4]. This was the reason that motivates the authors to name the architecture as the "K-filter".

* This work was supported by the National Research Plan of Spain, CICYT, Grant number TIC-92-0800-C05-05.

2. THE K-FILTER

Suppose an NLS with memory characterised by the Q-dimensional (Q-dim) input/output relation, g[.], and whose input signal, x(t) is limited to the range [-Xmax, Xmax].



$$y(t) = g[x(t), x(t-\tau), \dots, x(t-(Q-1)\tau)]$$

Fig.1.-Q-dim. input/output relation of a NLS.

This function and, therefore, the output of the NLS allows a Q-dim Fourier series developed not in the time domain but in the input domain as it is shown in equation (1). The coefficients of this series (Eq.2) coincide with the values of the Q-dim Fourier transform of g[.] at frequencies multiples of the so-called principal harmonics, ω_i . It is important to remark that in concordance with equation (1), the Fourier transform is also computed in the input domain. That is, the ith dimension of the Fourier transform corresponds to the transform of g[.] with respect to the variable $x(t-(i-1)\tau)$.

$$\hat{y}(t) = a_0 + \sum_{n_0=1}^{N_0} \sum_{n_1=1}^{N_1} \dots \sum_{n_{(Q-1)}=1}^{N_{(Q-1)}} \left\{ a_{n_0, n_1, \dots, n_{(Q-1)}} \cdot e^{-j[n_0 \omega_0 x(t) + n_1 \omega_1 x(t-\tau) + \dots + n_{(Q-1)} \omega_{(Q-1)} x(t-(Q-1)\tau)]} \right\} \quad (\text{Eq.1})$$

$$a_{n_0, n_1, \dots, n_{(Q-1)}} = F \left\{ g[x(t), \dots, x(t-(Q-1)\tau)] \right\}_{|n_0 \omega_0, n_1 \omega_1, \dots, n_{(Q-1)} \omega_{(Q-1)}} \quad (\text{Eq.2})$$

$$n_i = 1 \dots N_i \quad \omega_i \geq \frac{\pi}{X_{\max}} \quad \forall i = 0 \dots (Q-1)$$

As it has been mentioned previously, the K-filter consists in a truncated version of the Q-dim. Fourier series (Eq.1). It may be inferred from figure (2) that the K-filter keeps the coefficients from equation (1) that correspond to the axis of the Fourier transform, where only one frequency is different from zero. On the other hand, the K-filter takes N coefficients from each addition, which in general will be less than N_i . These two features make the K-filter to be a short version of the Q-dim Fourier series. Furthermore, the architecture performs a non-uniform sampling in the transformed domain. This process is included in the K-filter by adding a second multidimensional Fourier series with slightly different principal harmonics, $\omega_i' \forall i=0 \dots (Q-1)$, as it is shown in figure (2).

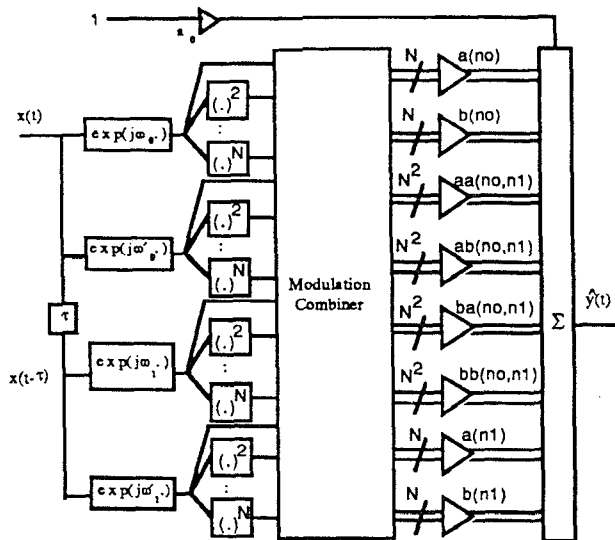


Fig.3.-The K-filter(2,N) with modulation combiner.

This new version of the K-filter (Fig.3) still agrees with the Fourier transform as it was explained in section 2. Actually it fits better the Q-dim Fourier series than the original K-filter (Fig.2). The reason lies in the fact that the K-filter with modulation combiner includes not only samples from the axis of the Fourier transform of $g[.]$ but also some new samples out of these axis. These new samples are generated by the function named modulation combiner since it multiplies harmonics from different branches or, what it is the same, from different axis of the Fourier transform. The effect of the K-filter and the K-filter with modulation combiner over the Fourier transform in a 2-dim case is shown in figure (4). Assuming a uniform sampling, the original K-filter takes only samples of the axis (dots), whereas the K-filter with modulation combiner samples also outside the axis (crosses) following a squares.

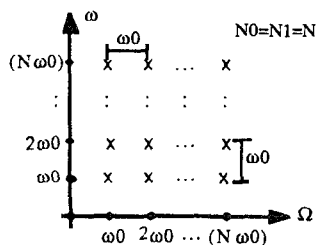


Fig.4. Samples taken by the K-filter (dots) and the K-filter with modulation combiner (dots&crosses) in a 2-dim. case.

Note that in a modelling problem where the input/output function of the NLS is known, the coefficients of the K-filter can be still obtained directly from the real values of sampling the Q-dim Fourier transform of that function. On the other hand, the K-filter with modulation combiner of figure (3) may be extended to any number of delays but note that the number of coefficients increases at the rate of N^Q . Then, the computational load that the resulting filter would require makes the scheme not feasible for large values of Q.

4. SIMULATION RESULTS

This section is devoted to show the performance of the K-filter in the adaptive identification of a NLS with memory that consists in the model of a communication system (Fig.5). The non-linearity of the system is included by the amplifiers located at the transmitter and receiver whose in/out function is $g[x]=\text{sign}(x) \exp(x^2/0.05)$.

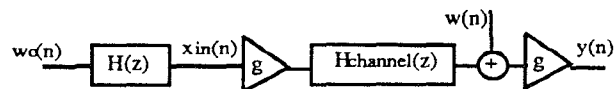


Fig.5. Model of a communication system.

The input, $x_{in}(n)$, has been selected as a sampled band-pass normal distributed noise filtered through the system, $H(z)=(z^{-4}+2.7607z^{-3}+3.8106z^{-2}+2.6535z^{-1}+0.9238)^{-1}$. On the other hand, the channel is modelled by the filter $H_{ch}(z)=(z^{-3}+0.0928z^{-2}-0.3158z^{-1}+0.2)/(z^{-1}-0.5)$. An additive gaussian noise of 10 dB has been also taken into account. It is important to remark that the amplifiers are memoryless but the channel includes the memory to the whole system. Then, we face with the problem of identifying a NLS with memory (Fig.6)

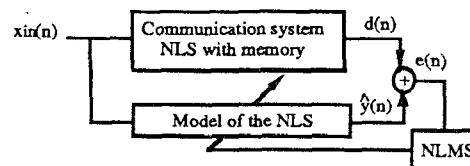


Fig.6. Identification problem.

The coefficients of the filter are modified with the NLMS adaptive algorithm (Eq.3). The \underline{u} variable is the data space vector, e is the error between the real output ($d(n)$) and the approximated one ($\hat{y}(t)$) and \underline{w} represents the coefficients vector. The μ variable is a real parameter that along with the instantaneous power of the data space build the variable step-size of the NLMS algorithm. Note that the instantaneous power of the data space, $(\|u(n)\|^2)$, in the K-filter is equal to the number of coefficients independently of the power of the input signal. This feature of the K-filter is due to the fact that each component of the data space vector is equal to a complex exponential.

$$w(n+1) = w(n) + \frac{\mu}{\|u(n)\|^2} e(n) \underline{u}(n) \quad (\text{Eq.3})$$

In order to compare the results four kind of filters have been chosen to model the whole non-linear communication system: first of all the K-filter (Q,N) with and without modulation combiner, then a FIR filter and finally a Volterra filter with memory (N,P) (Eq.4), being N the memory of the filter and P the order of the maximum Volterra functional that the filter includes.

$$y(t) = \sum_{p=0}^P h_p[x] \quad ; \quad x = [x(t), x(t-\tau), \dots, x(t-(N-1)\tau)]$$

$$h_p[x] = \sum_{n_1=0}^{N-1} \dots \sum_{n_p=0}^{N-1} h_{n_1, \dots, n_p} x(t-n_1\tau) \dots x(t-n_p\tau) \quad (\text{Eq.4})$$

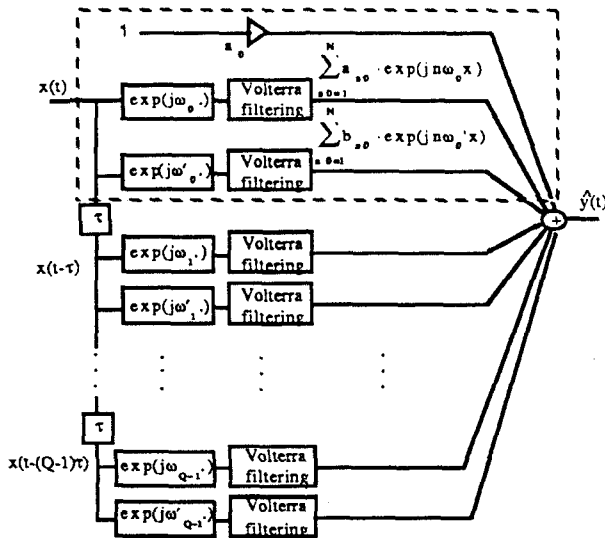


Fig.2.-The K-filter (Q,N): a model for NLSs with memory. The memoryless case is framed with a dashed line.

First of all we will focus our attention on the memoryless K-filter, framed with dashed line in figure (2). It implements a truncated Fourier series of the input/output memoryless relation $g[x(t)]$. Actually it performs two Fourier series with two different principal harmonics, ω_0 and ω'_0 , in order to obtain the nonuniform sampling of the transformed domain. This process is carried out by both branches supplied with the input signal, $x(t)$. Each branch consists of a FM modulator followed by an N-order memoryless Volterra series [5] which is another way of computing a Fourier series. Another third branch with unity input is also needed to fit the continuous component, since it is not included in the Volterra filtering.

As it may be inferred from figure (2), the K-filter for the case of NLSs with memory emerges as a generalisation of the memoryless architecture [4,6]. Taking into account that the Volterra filtering performs a memoryless series, the K-filter includes memory just by means of the input temporal diversity vector, $[x(t), x(t-\tau), \dots, x(t-(Q-1)\tau)]$. Each component is then used to compute a double Fourier series with principal harmonics ω_i and ω'_i in order to generate the nonuniform sampling in the corresponding axis of the Fourier transform.

In a modelling problem the number of delays (Q), the order of the Volterra filtering (N), the principal frequencies, ω_i and ω'_i , and the coefficients, (a_{ni}, b_{ni}) , are parameters of the K-filter that vary from one design to another. Whereas the number of delays is related to the memory of the NLS to be modelled, the rest of parameters depend on the Fourier transform of the input/output function, $g[\cdot]$. Note that the coefficients can be computed directly from sampling the Q-dim Fourier transform of $g[\cdot]$ whenever this function is available. On the other hand, in solving an identification problem, a minimum mean square error criterion will lead the K-filter to the Wiener solution since the output is a linear combination of the series coefficients. Therefore, all the adaptive algorithms that converge to the

Wiener solution as the gradient-based algorithms (LMS, NLMS) or recursive algorithms (RLS, SQRLS) can be applied to the K-filter as it will be shown in the next section.

It is also important to remark that the values of the frequencies, ω_i and ω'_i , determine the performance of the K-filter. First of all, we have to bear in mind that the K-filter takes samples in the transformed domain at multiples of the principal harmonics. Then, if any of these multiples does not match with significant values of the Fourier transform of $g[\cdot]$, the approximation that the K-filter may achieve will be poor. Furthermore, when the Wiener solution is computed in solving, for instance, an identification problem, the well-conditioned or not of the autocorrelation matrix is strongly affected by the value of this principal harmonics. This two features make the selection of the principal frequencies a crucial point in the design of the K-filter.

We may conclude that the proposed scheme allows an improvement of the results by increasing the order of the Volterra series and/or increasing the number of delays. Although these modifications will also increase the computational load of the K-filter, they can be necessary when certain accuracy is required.

3. NEW VERSIONS OF THE K-FILTER

Once arrived at this point, the new goal was not only to check the performance of the K-filter when modelling NLSs but also to develop more accurate versions of it in order to improve the results. From the point of view of the authors a possible way would consist in enlarging the architecture, making it at the same time more complicated but also more general. This section is devoted to present the results that the authors have arrived to. Although it is only a new version of the K-filter, in our opinion it is necessary in order to complete the previous scheme.

3.1. The K-Filter with modulation combiner.

When the K-filter (Q,N) is compared with the model built from a Q-th order Volterra functional [5], an important difference seems to make it less suitable to model certain NLSs than the other one. In fact, the K-filter does not include cross-products between different components of the input temporal diversity whereas the Volterra functional does. A possible way to overcome this disadvantage is to insert in the K-filter a function that combines variables from different branches. This function can be applied to the input temporal diversity vector just at the beginning of the K-filter, but also to the output of the exponential transformation or even to the output of the Volterra filtering before the last addition. From the point of view of the authors the best place is inside the Volterra filtering just before making the linear combination with the coefficients. By inserting the combination function in this place, the resulting filter can be still viewed as a Q-dim Fourier series. It also completes mathematically the original scheme, as it will be shown later. The new version is shown in figure (3) where only one delay has been considered in order to simplify the scheme.

The simulations show the evolution of the mean square error (MSE) with respect to the input power achieved by different models. Two different criteria have been considered in order to compare the results: the number of coefficients and the memory. Hence that figure (7) corresponds to filters with the same number of coefficients, whereas figure (8) shows the results achieved by different models that have the same number of components of the input temporal diversity vector, that is, the same memory. First of all, three different models with, approximately, the same number of coefficients have been checked: a Volterra filter (3,4) with $\mu=0.01$ (51 coef.), a FIR filter (37 coef.) with $\mu=0.1$, a K-filter (2,3) with $\mu=0.1$ and principal frequencies $\omega_0=\omega_1=\omega_2=\pi/(1.3X_{max})$ and $\omega'_0=\omega'_1=\omega'_2=3.66\omega_0$. Henceforth, the K-filter has 18 complex coefficients and one real. This is the reason to have chosen a 37-tap FIR filter. Even in this situation where the FIR filter has more temporal information than the K-filter, this last one performs better. Figure (7) also includes the result obtained from a K-filter (1,2) with modulation combiner .

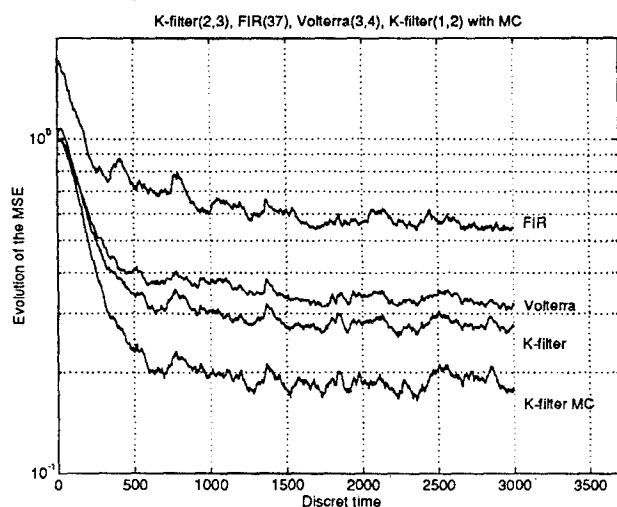


Fig. 7. Evolution of the mean square error (MSE).

The final value of the error of the FIR filter has an important misadjustment in relation to the minimum MSE obtained when the coefficients are computed off-line with the Wiener solution. That is, having a 1000 sampled input signal, the MSE of the FIR filter is 35%, of the K-filter is 25%, the Volterra filter gives 23% and the K-filter with modulation combiner has one of 15%. It is possible to think that the best performance of the K-filter is due to the fact that it has more coefficients (81). Hence, figure (8) shows the results obtained with a FIR filter with 81 coefficients, a Volterra filter (2,12) and the K-filter (1,2) with modulation combiner. Note that the Volterra filter includes the same temporal components and has approximately the same number of coefficients, 91. The results of a FIR filter with 3 coefficients have been also included.

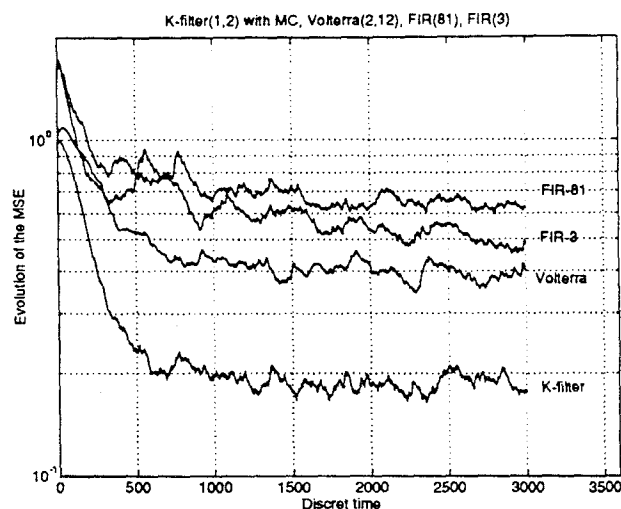


Fig. 8.- Evolution of the mean square error.

Figures (7,8) only want to show how the K-filter performs in the adaptive identification of a NLS with memory. Obviously, it does not avoid the inherent characteristics of the LMS algorithm as the trade-off between the step-size parameter and the misadjustment error, or the strong dependence on the spread of the eigenvalues. This problem can be solved by using another algorithms less sensible to these problems. At any rate, it is possible to infer that the K-filter with or without modulation combiner gives good chances when adaptive techniques are used.

REMARKS

A new architecture able to model NLSs both memoryless and with memory has been presented. This filter is named K-filter because it follows the main guide lines of the Kolmogorov's mapping theorem. It has been also deduced the K-filter with modulation combiner, a new version that completes the original scheme from the mathematical point of view. The performance of both versions have been checked in the adaptive identification of a NLS. Future works will be centered on studying the behaviour of the K-filter in new applications and on the possibility of simplifying it taking into account characteristics that NLSs usually share, without affecting the performance.

REFERENCES

- [1] J.S. Bendat: "Nonlinear system analysis & identification from random data", John Wiley & Sons, 1990.
- [2] S.Neil Rasband: "Chaotic dynamics of nonlinear systems", John Wiley & Sons, 1990.
- [3] I. Bilinskis, A. Mikelsons, "Randomised Signal Processing", Prentice Hall, 1992.
- [4] M.A. Lagunas: "The Kolmogorov mapping theorem in signal processing". Procc. 7SP Workshop on Statitstical Signal Array Processing, Quebec City, Canada June 26-29, 1994.
- [5] M. Schetzen: "The Volterra/Wiener theories of nonlinear systems", John Wiley & Sons, 1980.
- [6] A.Pagès-Zamora, M.A.Lagunas: "A novel architecture to model non-linear systems". Procc. 19SP ICASSP, Adelaide, Australia April 19-22, 1994.