Nonlinear Time Series Modeling and Prediction Using Functional Networks. Extracting Information Masked by Chaos

E. Castillo and J.M. Gutiérrez

Dpto. de Matemática Aplicada y Ciencias de la Computación, Universidad de Cantabria, E-39005, SPAIN
Physics Lett. A. **244***, 71-84 (1998).*

In functional networks, the topology of the network is a graphical representation of the properties we may know about the dynamics of the system.

The threshold-like nerual sigmoidal functions may be inappropriate to model the actual dynamics of non-linear time series. Therefore, in functional networks they are replaced by functions from appropriate families for each specific problem.

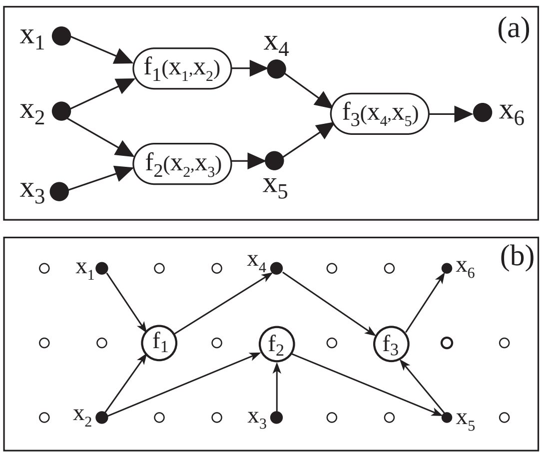

Fig. 1. (a) Functional network with three input, two intermediate, and one output storing units and three neuron, or processing, units; (b) equivalent parallel printed circuit board.

Preprint submitted to Elsevier Science *20 April 1999*

An interesting family of functional network architectures is the so called separable functional networks, which has associated a functional expression which combines the separate effects of input variables. For the case of two inputs, x and y, and one output, z, we have:

$$z = F(x, y) = \sum_{i=1}^{n} f_i(x) g_i(y). \tag{1}$$

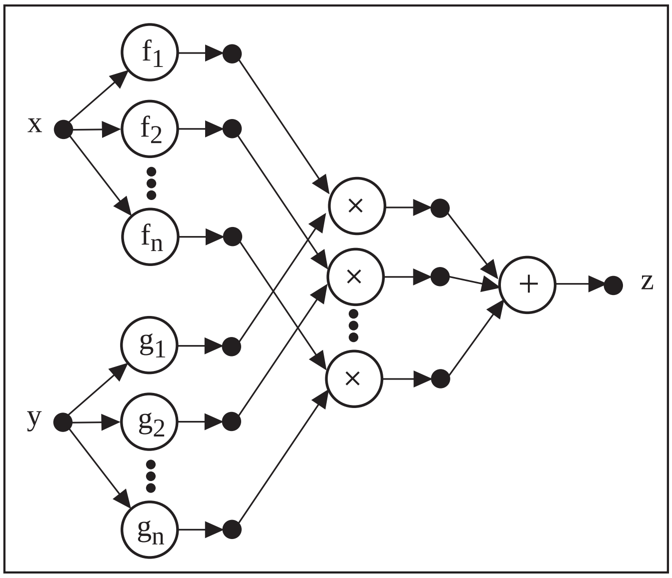

Fig. 2. Separable functional network architecture with two inputs and one output.

For illustrative purposes, we start by considering the simplest architecture from this family, which neglects double interactions by separating the contributions of each of the inputs in the form

$$z = F(x, y) = f(x) + g(y). \tag{2}$$

Uniqueness of representation !!!

$$F(x,y) = f_1(x) + g_2(y) = f_1^*(x) + g_2^*(y). \tag{3}$$

Theorem 1 *All solutions of equation* $\Sigma_{i=1}^n f_i(x)g_i(y) = 0$ *can be written in the form* $f(x) = A\,\varphi(x)$, $g(y) = B\,\psi(y)$, *where* A *and* B *are constant matrices (of dimensions* $n \times r$ *and* $n \times n - r$, *respectively) with* $A^T B = 0$, *and* $\varphi(x) = (\varphi_1(x), ..., \varphi_r(x))$ *and* $\psi(y) = (\psi_{r+1}(y), ..., \psi_n(y))$ *are two arbitrary systems of mutually linearly independent functions, and* r *is an integer between* 0 *and* n.

$$\begin{pmatrix} f_1(x) - f_1^*(x) \\ 1 \end{pmatrix} = \begin{pmatrix} c_1 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 \\ g_2(y) - g_2^*(y) \end{pmatrix} = \begin{pmatrix} 1 \\ c_2 \end{pmatrix}$$

where

$$\begin{pmatrix} c_1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ c_2 \end{pmatrix} = 0 \quad \Leftrightarrow \quad c_1 = -c_2 = c. \tag{4}$$

Finally, we get the constraints:

$$f_1^*(x) = f_1(x) - c, \; g_2^*(y) = g_2(y) + c, \tag{5}$$

where c is an arbitrary constant.

It is necessary to give **an initial value !** for one of the functions in order to eliminate the arbitrariness associated with constant c.

When the data is given in the form of a time series $\{x_k\}$ consisting of n points, we use an embedding of the time series in an appropriate delayed-coordinates space to train the functional network.

$$\{(x_{0i}, x_{1i}, x_{2i}); i = 3, \ldots, n\}$$

where: $x_{0i} = x_i$, $x_{1i} = x_{i-1}$, and $x_{2i} = x_{i-2}$.

Then we can approximate the functions f and g in (2) by considering a linear combination of known functions from a given family (in this paper we shall consider polynomials or Fourier expansions):

$$\hat{f}(x) = \sum_{j=1}^{m_1} a_{1j}\phi_{1j}(x), \quad \hat{g}(x) = \sum_{j=1}^{m_2} a_{2j}\phi_{2j}(x),$$

where the coefficients a_{kj} are the parameters of the functional network, i.e., they play the role of the weights on a neural network. Then, the error can be measured by

$$e_i = x_{0i} - \hat{f}(x_{1i}) - \hat{g}(x_{2i});\ i = 1, \ldots, n. \tag{6}$$

Thus, to find the optimum coefficients we minimize the sum of square errors

$$Q = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(x_{0i} - \sum_{k=1}^{2} \sum_{j=1}^{m_k} a_{kj}\phi_{kj}(x_{ki}) \right)^2 . \tag{7}$$

$$\hat{f}(u_0) \equiv \sum_{j=1}^{m_1} a_{1j}\phi_{1j}(u_0) = v_0, \tag{8}$$

where u_0 and v_0 are given constants.

$$
\begin{cases}
\dfrac{\partial Q_\lambda}{\partial a_{1r}} = -2\sum\limits_{i=1}^{n}\left(x_{0i} - \sum\limits_{k=1}^{2}\sum\limits_{j=1}^{m_k} a_{kj}\phi_{kj}(x_{ki})\right)\phi_{1r}(x_{1i}) + \lambda\phi_{1r}(u_0) = 0, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad r = 1, \ldots, m_1, \\
\dfrac{\partial Q_\lambda}{\partial a_{2r}} = -2\sum\limits_{i=1}^{n}\left(x_{0i} - \sum\limits_{k=1}^{2}\sum\limits_{j=1}^{m_k} a_{kj}\phi_{kj}(x_{ki})\right)\phi_{2r}(x_{2i}) = 0, r = 1, \ldots, m_2, \\
\dfrac{\partial Q_\lambda}{\partial \lambda} = \sum\limits_{j=1}^{m_1} a_{1j}\phi_{1j}(u_0) - v_0 = 0.
\end{cases}
\tag{9}
$$

Inferring Nonlinear Models From Time Series

As a first example, we consider the Hénon map:

$$
x_n = 1 + y_{n-1} - a x_{n-1}^2, y_n = 0.3 x_{n-1}, \tag{10}
$$

or, using delayed coordinates, as

$$
x_n = 1 - a x_{n-1}^2 + 0.3 x_{n-2}. \tag{11}
$$

The Holmes map is a cubic 2D map that can be written as:

$$
x_n = 2.76\, x_{n-1} - x_{n-1}^3 - 0.2\, x_{n-2}, \tag{12}
$$

whereas the Lozi map involves non-differentiable functions which make difficult the modeling of the associated time series. This map is given by:

$$
x_n = 1 - 1.7\, |x_{n-1}| + 0.5\, x_{n-2}. \tag{13}
$$

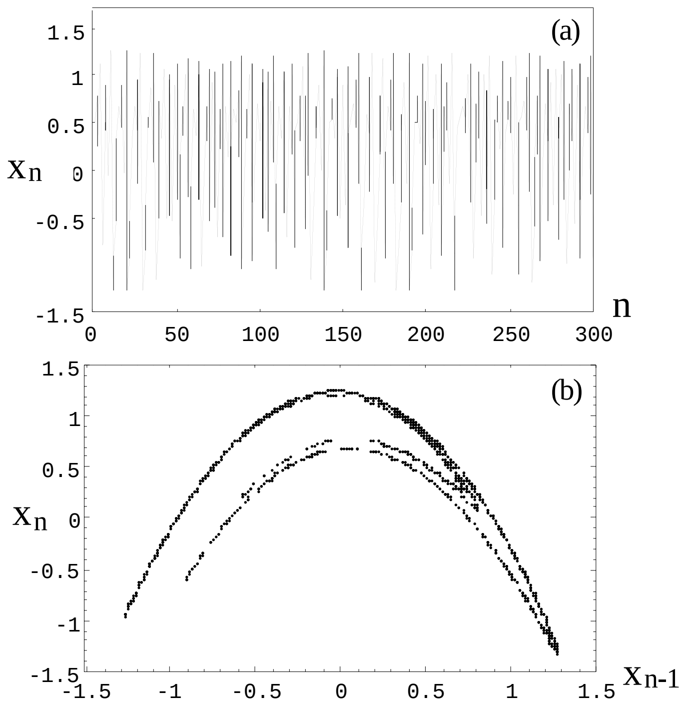

Fig. 3. (a) Time series of a chaotic orbit of the Hénon map. (b) Phase space of the first embedding of the system showing the quadratic relationship between the variables.

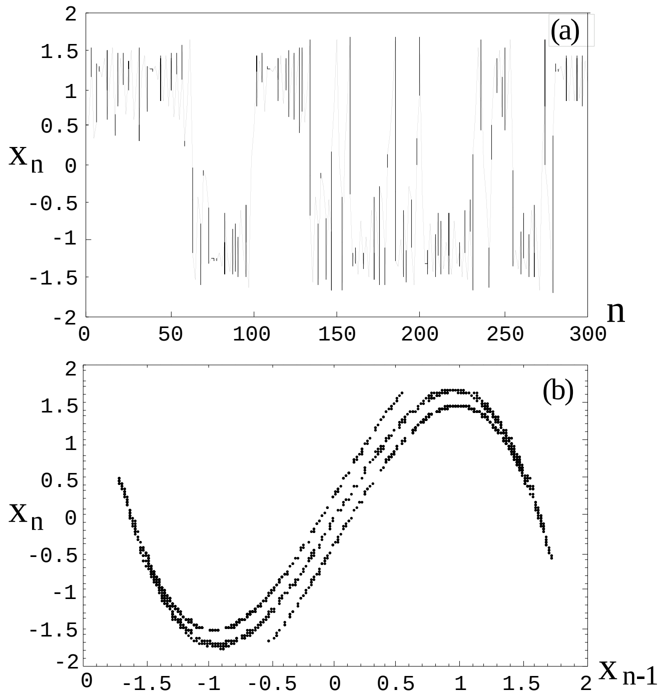

Fig. 4. (a) Time series of a chaotic orbit of the Holmes map. (b) Phase space of the first embedding of the system showing the cubic relationship between the variables. The initial conditions are $x_0 = 0.1$ and $x_1 = 0.3$.

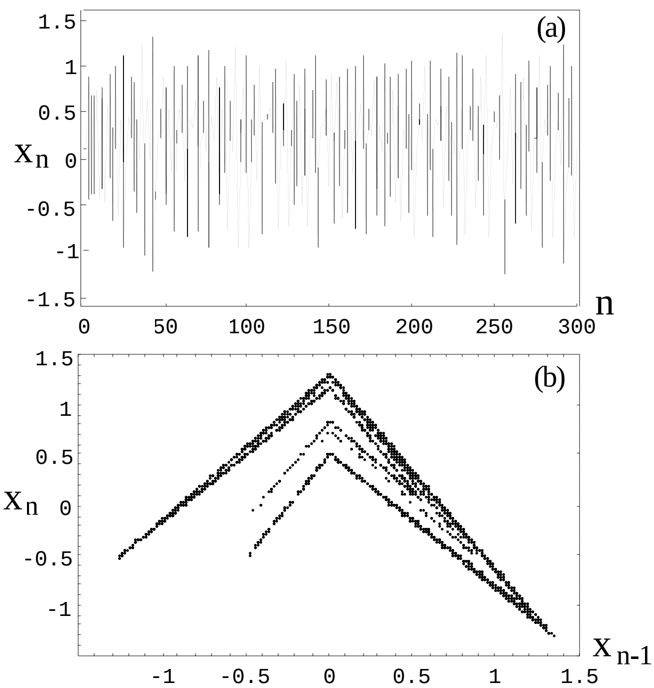

Fig. 5. (a) Time series of a chaotic orbit of the Lozi map. (b) Phase space of the first embedding of the system showing the non-differentiable relationship between the variables. The initial conditions are $x_0 = 0.5$ and $x_1 = 0.7$.

Network	Par.	RMSE Training Data			RMSE Test Data		
		Henon	Holmes	Lozi	Henon	Holmes	Lozi
m=4	16	0.0058	0.0099	0.038	0.0064	0.026	0.039
m=5	20	$7.8\ 10^{-4}$	0.0023	0.028	$7.9\ 10^{-4}$	0.0061	0.028
m=6	24	$4.5\ 10^{-5}$	$3.8\ 10^{-4}$	0.021	$9.3\ 10^{-5}$	$9.9\ 10^{-4}$	0.025
m=7	28	$6.7\ 10^{-6}$	$8.3\ 10^{-5}$	0.016	$2.2\ 10^{-5}$	$4.1\ 10^{-4}$	0.02

Table 1
Performance of several Fourier functional networks for the Hénon, Holmes and Lozi time series. The number of parameters and the RMS errors obtained in each case are shown.

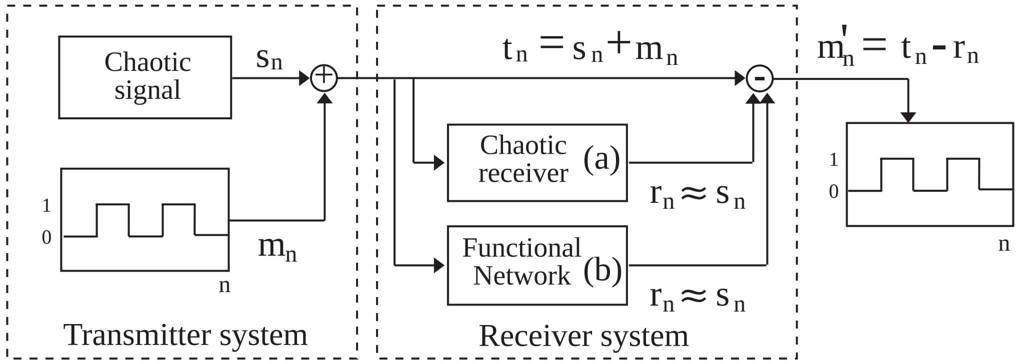

Fig. 6. Scheme for secure communications based on chaos synchronization.

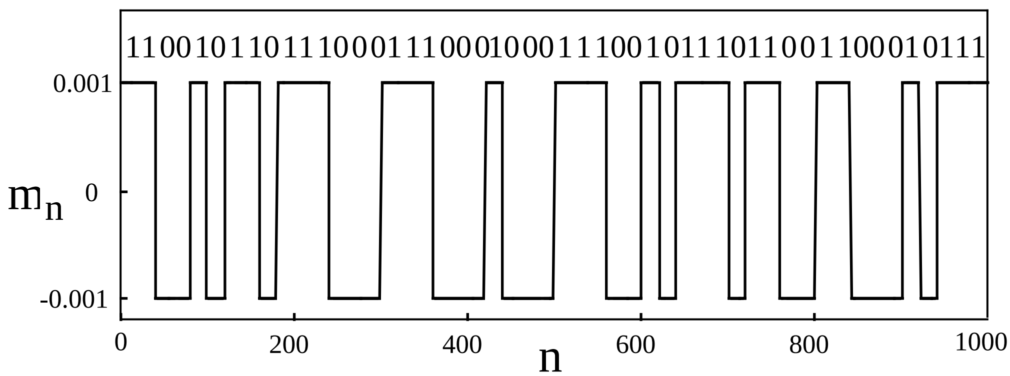

Fig. 7. Binary message "110010110111..." where each bit is represented by 20 consecutive sequence points.

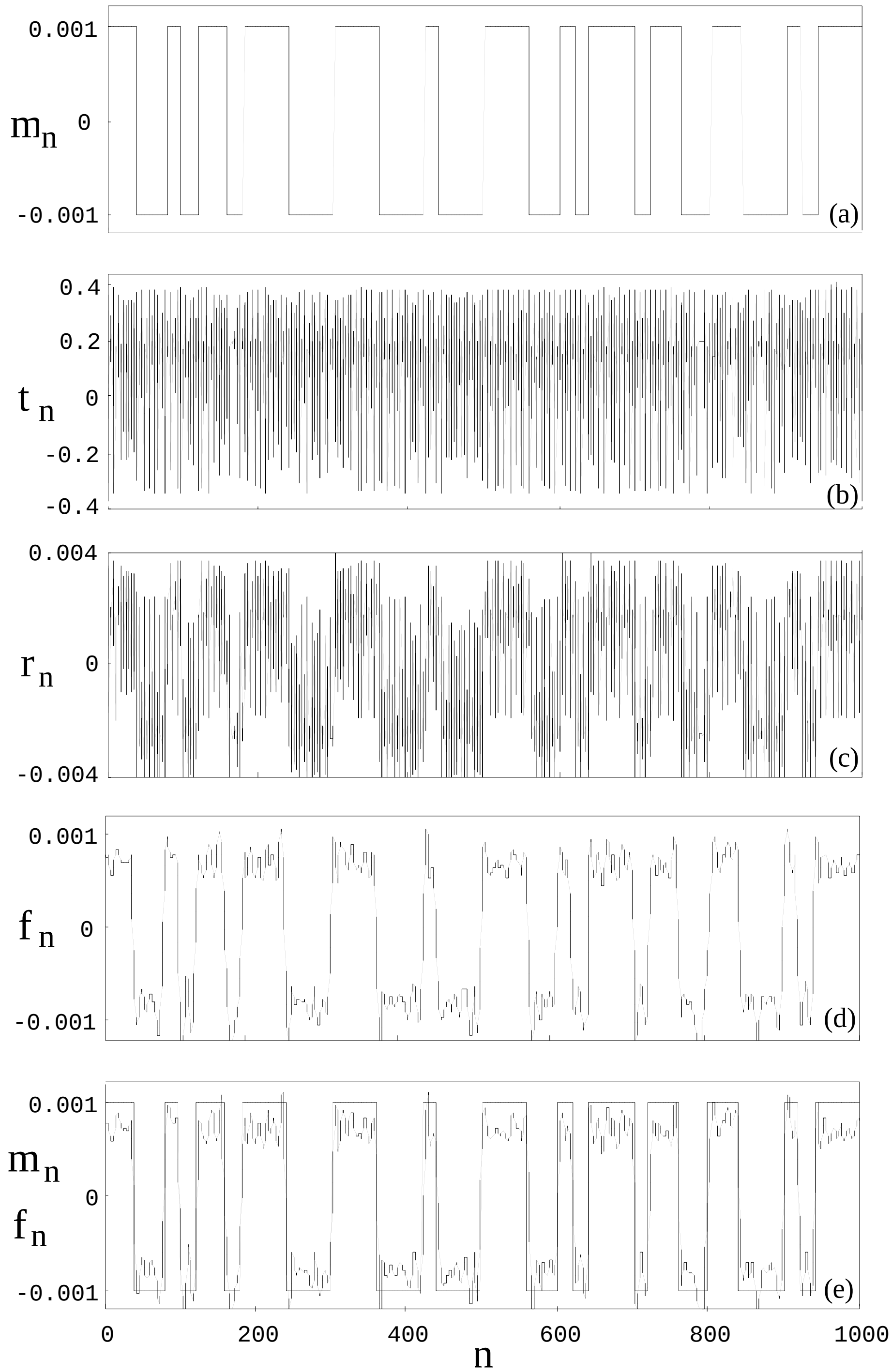

Fig. 8. Unmasking a digital signal transmission: (a) the original binary m_n-message where each bit is represented by 20 points of the series m_n; (b) actual chaotic transmitted signal (message + chaotic signals); (c) reconstructed message obtained from the exact polynomial functional network in (1); (d) low-pass filtered signal of r_n. Finally, (e) shows both the original and recovered messages.

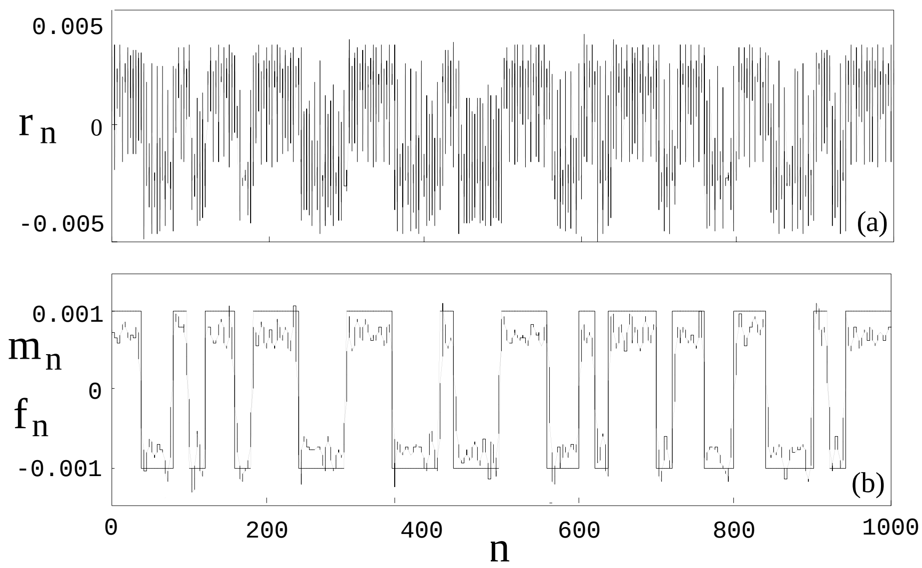

Fig. 9. Unmasking a digital signal transmission: (a) reconstructed message obtained from the Fourier approximate functional network with $m = 7$. (b) Low-pass filtered received and original messages.