Learning time series evolution by unsupervised extraction of correlations

Gustavo Deco* and Bernd Schürmann
Siemens AG, Corporate Research and Development, ZFE ST SN 41, Otto-Hahn-Ring 6, 81739 Munich, Germany
(Received 29 March 1994; revised manuscript received 29 August 1994)

We focus on the problem of modeling time series by learning statistical correlations between the past and present elements of the series in an unsupervised fashion. This kind of correlation is, in general, nonlinear, especially in the chaotic domain. Therefore the learning algorithm should be able to extract statistical correlations, i.e., higher-order correlations between the elements of the time signal. This problem can be viewed as a special case of factorial learning. Factorial learning may be formulated as an unsupervised redundancy reduction between the output components of a transformation that conserves the transmitted information. An information-theoretic-based architecture and learning paradigm are introduced. The neural architecture has only one layer and a triangular structure in order to transform elements by observing only the past and to conserve the volume. In this fashion, a transformation that guarantees transmission of information without loss is formulated. The learning rule decorrelates the output components of the network. Two methods are used: higher-order decorrelation by explicit evaluation of higher-order cumulants of the output distributions, and minimization of the sum of entropies of each output component in order to minimize the mutual information between them, assuming that the entropies have an upper bound given by Gibbs second theorem. After decorrelation between the output components, the correlation between the elements of the time series can be extracted by analyzing the trained neural architecture. As a consequence, we are able to model chaotic and nonchaotic time series. Furthermore, one critical point in modeling time series is the determination of the dimension of the embedding vector used, i.e., the number of components of the past that are needed to predict the future. With this method we can detect the embedding dimension by extracting the influence of the past on the future, i.e., the correlation of remote past and future. Optimal embedding dimensions are obtained for the Hénon map and the Mackey-Glass series. When noisy data corrupted by colored noise are used, a model is still possible. The noise will then be decorrelated by the network. In the case of modeling a chemical reaction, the most natural architecture that conserves the volume is a symplectic network which describes a system that conserves the entropy and therefore the transmitted information.

PACS number(s): 02.50.−r, 89.70.+c, 05.45.+b

I. INTRODUCTION

Modeling time series by learning from experiments can be viewed as the extraction of statistical correlations between the past and future values of the time series signals. In particular, in the case of chaotic series, due to their short-term predictability, a thorough study of statistical correlations between components of the embedding vector yields the only way to distinguish between a purely random process and a chaotic deterministic series, eventually corrupted by colored or white noise. In fact, most of the relevant dynamic invariants that characterize such series are measures of these correlations between past values and the future evolution of the time series. The two most important of these are the largest Lyapunov exponent which contains information on how far in the future predictions are possible, i.e., how far the future is statistically correlated with the past [1], and the correlation dimension which measures the information contained in the attractor [2]. Long-term forecasting of the behavior of chaotic systems is not possible due to the exponential divergence of trajectories in such systems [3].

Several authors [4–8] proposed a great variety of nonneural models that yield very accurate results for short-term prediction. Neural network modelings were also implemented using supervised learning paradigms (i.e., paradigms for learning with a teacher) and feedforward [9–13] or recurrent architectures [14]. But the problem of extracting statistical correlations in a sensorial environment is the subject of unsupervised learning (i.e., the only available information is in the correlations of the input data, and no teacher is present). In fact, Barlow [15,16] proposed the principle of redundancy reduction as the goal of unsupervised learning. The brain performs statistical decorrelation of the input environment in order to extract statistically independent relevant information. The goal of redundancy reduction is to factorize the input probability distribution without loosing information. In the linear case Barlow's principle yields a learning rule that performs a principal component analysis (PCA). In fact, PCA can be derived as a linear transformation which conserves transmission of information and minimizes the mutual information between the outputs in or-

*Electronic address: deco@inf21.zfe.siemens.de
FAX: +49 89 636 3320.

der to decorrelate them. Some nonlinear extensions of PCA for decorrelation of sensorial input signals were recently introduced [17–20]. Atick and Redlich [17,18] and Redlich [19,20] concentrate on the original idea of Barlow obtaining a very interesting formulation of early visual processing and factorial learning. Redlich [20] reduces redundancy at the input by using a network structure which is a reversible cellular automaton and therefore guarantees the conservation of information in the transformation between input and output.

The aim of the present work is to formulate an architecture and a learning paradigm for the unsupervised extraction of statistical correlations between the past and future elements of a time signal in order to model the behavior of the dynamical system. This model performs Barlow's unsupervised learning in the most general fashion. In other words, we implement a *nonlinear principal component analysis* for the extraction of causal statistical correlations between the past and future of a time signal obtained from observations on a chaotic dynamical system. In this form, we extend the standard statistical techniques (e.g., PCA) which are applied for the modeling of dynamical systems.

We employ a single layer architecture that attempts to extract correlations considering only the past relative to each element of an embedding vector. The architecture is always reversible, conserves the volume and therefore the transmitted information. In general, the environment is non-Gaussian distributed and nonlinearly correlated. The learning rule decorrelates statistically the elements of the output by two different methods: statistical decorrelation by consideration of higher-order cumulants, or minimization of an upper bound of the mutual information between the components of the output using Gibbs second theorem. In the case where a coupled system of chemical reactions is modeled, i.e., equations are extracted from data, a symplectic architecture for the decorrelation between fluxes and concentration is employed. This architecture is a special case of an information conserving network. Put differently, we apply the herein introduced generalization of factorial learning of nonlinear and non-Gaussian environments to the special case of dynamic modeling of chaotic time series or dynamical chemical systems. In the case of chaotic modeling the method yields an information-theoretic-based theory for the determination of the embedding dimension which is a very actual problem in modern chaos research (see Refs. [21–24]). In the modeling of chemical systems, the method yields a way to extract the chemical kinetics of the participant elements by learning from data. The theory is illustrated by various examples.

II. DYNAMICAL MODELING

Before formulating the model we introduce in this section the basic assumption for the modeling of deterministic chaotic systems and coupled chemical reactions.

A. Deterministic chaos

For modeling a chaotic system from the observation of measures realized on the chaotic attractor we review briefly the Takens methods [25] which is called *phase-space reconstruction,* and results in a d-dimensional "embedding space" in which the dynamics of the multidimensional attractor is captured.

Let us assume a time series of a single (one-dimensional) measured variable of a multidimensional dynamical system. The aim of forecasting is to predict the future evolution of this variable. It has been shown that in nonlinear deterministic chaotic systems it is possible to determine the dynamical invariants and the geometric structure of the many-variable dynamic system that produces the single measurement [25,26] from the observation of a single dynamical variable.

Given is a chaotic system

$$\vec{y}(t+1)=\vec{g}(\vec{y}(t)) \ . \tag{1}$$

Let us define an observable measurement

$$x(t)=f(\vec{y}(t)) \ . \tag{2}$$

The Takens theorem assures that for an embedding

$$\vec{\xi}(t)=(x(t),x(t-\tau),\ldots,x(t-d\tau)) \tag{3}$$

a map

$$\vec{\xi}(t+1)=\vec{F}(\vec{\xi}(t)) \tag{4}$$

exists which has the same dynamical characteristics as the original system $\vec{y}(t)$ if $d=2D+1$ where D is the dimension of the strange attractor. This sufficient condition may be relaxed to $d>2D$ [25]. The theorem implies that all the coordinate-independent properties of $\vec{g}(\ )$ and $\vec{F}(\ )$ will be identical. The proper choice of d and τ is an important topic of investigation [21–24]. The goal of unsupervised neural network modeling is to learn the map given by $\vec{F}(\ )$ in Eq. (4), by learning the statistical correlations between the time successive elements (future from the past) of the embedding vector by observing the different training embedding patterns.

It is important to remark that the Takens theorem offers a method for modeling noise-free chaotical time series. When noise is added the embedding dimension method has serious problems (see, for example, Refs. [27] and [28]). In this paper, however, we used in one example a chaotic system (logistic map) perturbated with colored noise, and we have obtained a good result in the sense that the correlation between the embedding coordinates were extracted in spite of the noise, which was also decorrelated.

B. Chemical kinetic equations

In this case we would like to model a nonchaotic system of coupled first-order differential equations. Typically the form of the system is binomial, i.e.,

$$\frac{\partial \vec{x}}{\partial t}=\vec{k}_1+K_2\vec{x}(t)+\vec{x}_{(t)}\otimes K\otimes\vec{x}_{(t)} \ , \tag{5}$$

where $\vec{x}(t)$ is a column vector whose elements are the concentrations of the different reactants at time t. The vectors $\vec{k}_1$, the matrices K_2, and the third-order tensor K contain the kinetics of the chemical system. The symbol

$\otimes$ means tensorial product. The goal of learning in this case is to find the kinetic constants, and which reactions explain the experiments, i.e., the kinetics of the stoichiometry of the chemical system. The data consist of the time evolution of the concentration vector $\vec{x}(t)$ (measured concentrations).

III. UNSUPERVISED EXTRACTION OF STATISTICAL CORRELATIONS

In this section we present the architecture and the learning paradigm that perform an unsupervised extraction of correlations in the input given by the embedding vector of a time series by reducing the redundancy between the extracted features and conserving the information. Let us define an input vector $\vec{x}$ of dimension d distributed according to the probability distribution $p(\vec{x})$ which is not factorial, i.e., the components of $\vec{x}$ are correlated. The goal of Barlow's unsupervised learning rule is to find a transformation

$$\vec{y}=\vec{F}(\vec{x}) \tag{6}$$

such that the components of the d-dimensional output vector $\vec{y}$ are statistically decorrelated, i.e.,

$$P(\vec{y})=\prod_{i}^{d} P(y_i) , \tag{7}$$

and the information is transmitted without loss. The transformation $\vec{F}$ will be performed by a neural network. The constraint of perfect transmission of information for deterministic neural networks is equivalent to the condition of invertibility of the network. We use in this work information theory for the formal presentation of the model. Let us define a measure of the transmission of information, i.e., the mutual information $H(\vec{y};\vec{x})$ between input and output should be [29]

$$H(\vec{y};\vec{x})=H(\vec{x})-H(\vec{x}/\vec{y}) , \tag{8}$$

where the symbol $H(\vec{a})$ indicates the Shannon entropy [29] of $\vec{a}$, and $H(\vec{a}/\vec{b})$ the conditional entropy of $\vec{a}$, given by $\vec{b}$. The transmitted entropy satisfies

$$H(\vec{y})\leq H(\vec{x})+\int P(\vec{x}) \ln \left[\det \left[\frac{\partial \vec{F}}{\partial \vec{x}} \right] \right] d\vec{x} , \tag{9}$$

where equality holds only if $\vec{F}$ is bijective, i.e., reversible. Conservation of information and bijectivity is assured if the neural transformation conserves the volume, which mathematically can be expressed by the fact that the Jacobian of the transformation should have determinant unity, i.e.,

$$\det \left[\frac{\partial \vec{F}}{\partial \vec{x}} \right] = 1 , \tag{10}$$

where $\partial \vec{F}/\partial \vec{x}$ is the Jacobian matrix of the neural network transformation. We formulate an architecture that conserves always the volume, independent of the values of its weights. Using (10) we can reduce (9) to

$$H(\vec{y})=H(\vec{x}) . \tag{11}$$

The architecture employed in this paper is shown in Fig. 1(a). The dimensions of input and output layer are the same and equal to d. Redlich [20] proposed this kind of architecture using the theory of reversible cellular automata. The analytical definition of the transformation defined by this architecture can be written as

$$y_i=x_i+f_i(x_0,\ldots,x_j,\vec{\omega}_i) \quad \text{with } j<i \tag{12}$$

where $\vec{\omega}_i$ represents a set of parameters of the function f_i. Note that independent of the functions f_i, the network is always volume conserving and satisfies Eq. (10). In particular, f_i can be calculated by another neutral network, by a sigmoid neuron, by polynomials (higher-order neurons), etc. Due to the asymmetric dependence on the input variables and the direct connections with weights equal to one between corresponding components of input and output neurons, the Jacobian matrix of the transformation in Eq. (12) is an upper triangular matrix with diagonal elements all equal to one, yielding a determinant equal to one. In this paper we will use higher-order networks. The functions f_i are assumed to be polynomial. The outputs are given by the following update equations:

$$y_i=x_i+\sum_{j=0}^{i-1}\omega_{ij}x_j+\sum_{j,k=0}^{i-1}\omega_{ijk}x_k x_j+\cdots . \tag{13}$$

The triangular structure of this network not only assures conservation of entropy in the transmission from the inputs to the outputs but also a transformation that attempts to decorrelate a component from only the past components which is the kind of correlation that we need in time series modeling.

Figure 1(b) shows another volume-conserving architec-

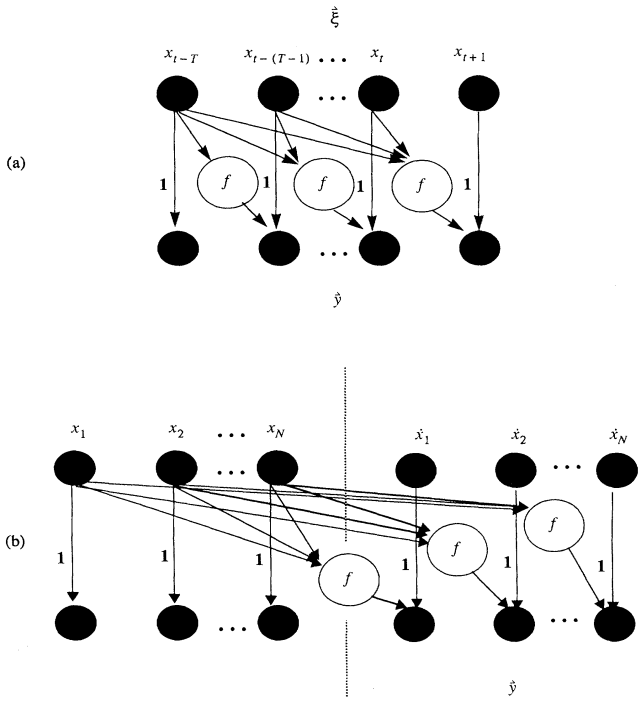

FIG. 1. (a) Volume-conserving neural architectures. (b) Symplectic neural architecture for flux modeling of coupled chemical reactions.

ture convenient for modeling a system of coupled chemical equations as given by Eq. (5). In this case only one point in the past is necessary for each concentration. The connections included are sufficient for building the map of the flux. The flux is obtained as a function of only the past value of the concentrations. The network of Fig. 1(b) is not only volume conserving but also "symplectic" [30]. A symplectic transformation satisfies the following equation:

$$S_N = \left[\frac{\partial \vec{F}}{\partial \vec{x}}\right]^T S_N \left[\frac{\partial \vec{F}}{\partial \vec{x}}\right] , \tag{14}$$

where

$$S_N = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \tag{15}$$

and I is the unity matrix. By using Eq. (14) it is easy to prove that the composition of symplectic transformations is also symplectic. The Jacobian associated with the architecture of Fig. 1(b) has the form

$$\left[\frac{\partial \vec{F}}{\partial \vec{x}}\right] = \begin{bmatrix} I & W \\ 0 & I \end{bmatrix} \tag{16}$$

which satisfies (14). The fact that this architecture conserves the information has its correspondence in statistical mechanics in the conservation of entropy for conservative systems. The update equations are given by

$$y_i = x_i , \quad i = 1, \ldots N \tag{17}$$

$$y_i = \dot{x}_i + \sum_{j=0}^{N} \omega_{ij} x_j + \sum_{j,k=0}^{N} \omega_{ijk} x_k x_j + \ldots ,$$

$$i = N+1, \ldots, 2N \tag{18}$$

where $\dot{x}_i = \partial x_i / \partial t$ and y_i is the output of the network. After training, the outputs y_i, $i = N+1, \ldots, 2N$ should be constants, which indicates decorrelation between the concentration x_j and the flux $\dot{x}_i$.

Let us now concentrate on the second aspect of factorial learning, namely the decorrelation of the output components. Here the problem is to find an invertible transformation that satisfies Eq. (7). The major problem is that the distribution of the output signal is not necessarily Gaussian.

We now present two techniques to achieve this goal. The first consists of the decorrelation of non-Gaussian distributions by analyzing the decomposition of higher-order cumulants. The second consists of minimizing the mutual information between the components of the output. In this last case we use the second theorem of Gibbs for minimizing an upper bound.

A. Cumulant expansion

In order to decorrelate non-Gaussian distributions we expand the output distribution in higher orders of the correlation matrix and impose the independence condition (7). In order to achieve this we propose to use a cumulant expansion of the output distribution.

Let us define the Fourier transform of the output distribution,

$$\phi(\vec{K}) = \int d\vec{y}\, e^{i(\vec{K}\cdot\vec{y})} P(\vec{y}) , \tag{19}$$

$$\phi(K_i) = \int dy_i e^{i(K_i y_i)} P(y_i) . \tag{20}$$

The cumulant expansion of a distribution is [31]

$$\phi(\vec{K}) = \exp\left[\sum_{n=1}^{\infty} \frac{i^n}{n!} \sum_{i_1, i_2, \ldots, i_n}^{d} \aleph_{i_1, \ldots, i_n} K_{i_1} K_{i_2} \cdots K_{i_n}\right] , \tag{21}$$

$$\phi(K_i) = \exp\left[\sum_{n=1}^{\infty} \frac{i^n}{n!} \aleph_i^{(n)} K_i^n\right] . \tag{22}$$

The cumulants $\aleph_i^{(n)}$ and $\aleph_{i_1, \ldots, i_n}$ of Eqs. (21) and (22) are given in Ref. [31].

In Fourier space, the independence condition is given by [31]

$$\phi(\vec{K}) = \prod_i \phi(K_i) \tag{23}$$

which is equivalent to

$$\ln(\phi(\vec{K})) = \sum_i \ln\phi(K_i) . \tag{24}$$

Putting (24) and the cumulant expansions of (21) and (22) together we obtain that in the case of independence the following equality is satisfied:

$$\sum_{n=1}^{\infty} \frac{i^n}{n!} \sum_{i_1, i_2, \ldots, i_n}^{d} \aleph_{i_1, \ldots, i_n} K_{i_1} K_{i_2} \cdots K_{i_n} = \sum_i^d \left[\sum_{n=1}^{\infty} \frac{i^n}{n!} \aleph_i^{(n)} K_i^n\right] . \tag{25}$$

The first four multidimensional cumulants are given by

$$\aleph_i = \bar{y}_i , \tag{26}$$

$$\aleph_{ij} = C_{ij} - \bar{y}_i \bar{y}_j , \tag{27}$$

$$\aleph_{ijk} = C_{ijk} - C_{ij}\bar{y}_k - C_{jk}\bar{y}_i - C_{ik}\bar{y}_j + 2\bar{y}_i\bar{y}_j\bar{y}_k , \tag{28}$$

$$\begin{aligned} \aleph_{ijkl} = {} & C_{ijkl} - C_{ijk}\bar{y}_l - C_{ijl}\bar{y}_k - C_{ilk}\bar{y}_j \\ & - C_{ljk}\bar{y}_i - C_{ij}C_{kl} - C_{ik}C_{lj} - C_{il}C_{jk} \\ & + C_{ij}\bar{y}_k\bar{y}_l + C_{ik}\bar{y}_j\bar{y}_l + C_{il}\bar{y}_j\bar{y}_k + C_{jk}\bar{y}_i\bar{y}_l \\ & + C_{jl}\bar{y}_k\bar{y}_i + C_{kl}\bar{y}_i\bar{y}_j - 6\bar{y}_i\bar{y}_j\bar{y}_k\bar{y}_l , \end{aligned} \tag{29}$$

where

$$\bar{y}_i = \int dy_i y_i P(y_i) . \tag{30}$$

The first four one-dimensional cumulants can be derived similarly or can be taken from Ref. [31]. They are the following:

$$\aleph_i^{(1)} = \bar{y}_i , \tag{31}$$

$$\aleph_i^{(2)} = C_i^{(2)} - \bar{y}_i^2 , \tag{32}$$

$$\aleph_i^{(3)} = C_i^{(3)} - 3C_i^{(2)}\bar{y}_i + 2\bar{y}_i^3 , \tag{33}$$

$$\aleph_i^{(4)}=C_i^{(4)}-3(C_i^{(2)})^2-4C_i^{(3)}\bar{y}_i+12C_i^{(2)}\bar{y}_i^2-6\bar{y}_i^4 , \quad (34)$$

where the multidimensional higher-order moments $C_{i\dots j}$ are defined as

$$C_{i\dots j}=\int d\vec{y}'(P(\vec{y}')y_i'\cdots y_j' , \quad (35)$$

and the one-dimensional higher-order moments $C_i^{(n)}$ are given by

$$C_i^{(n)}=\int dy_i'P(y_i')(y_i')^n . \quad (36)$$

In order to remove the bias $\overline{(\vec{y})}$ we perform an extra transformation

$$\vec{y}'=\vec{y}-\overline{(\vec{y})} . \quad (37)$$

We may rewrite Eq. (25) using the cumulants expression:

$$-\tfrac{1}{2}\sum_{i,j}K_iK_j\{C_{ij}-C_i^{(2)}\delta_{ij}\}-\frac{i}{6}\sum_{i,j,k}K_iK_jK_k\{C_{ijk}-C_i^{(3)}\delta_{ijk}\}$$
$$+\tfrac{1}{24}\sum_{i,j,k,l}K_iK_jK_kK_l\{(C_{ijkl}-3C_{ij}C_{kl})-[C_i^{(4)}-3(C_i^{(2)})^2]\delta_{ijkl}\}=0 . \quad (38)$$

The $\delta_{i\dots j}$ denote Kroenecker's delta. Due to the fact that Eq. (38) should be satisfied for all $\vec{K}$, all coefficients in each summation must be zero. This means that

$$C_{ij}-C_i^{(2)}\delta_{ij}=0 , \quad \forall i,j \quad (39)$$

$$C_{ijk}-C_i^{(3)}\delta_{ijk}=0 , \quad \forall i,j,k \quad (40)$$

$$(C_{ijkl}-3C_{ij}C_{kl})-[C_i^{(4)}-3(C_i^{(2)})^2]\delta_{ijkl}=0 , \quad \forall i,j,k,l \quad (41)$$

or, equivalently,

$$C_{ij}=0 , \quad \text{if } (i\neq j) \quad (42)$$

$$C_{ijk}=0 , \quad \text{if } (i\neq j\vee i\neq k) \quad (43)$$

$$C_{ijkl}=0 , \quad \text{if } (\{i\neq j\vee i\neq k\vee i\neq l\}\wedge\neg L) \quad (44)$$

$$C_{iijj}-C_{ii}C_{jj}=0 , \quad \text{if } (i\neq j) . \quad (45)$$

In (44) l is the logical expression

$$L=\{(i=j\wedge k=l\wedge j\neq k)\vee(i=k\wedge j=l\wedge i\neq j)\vee(i=l\wedge j=k\wedge i\neq j)\} \quad (46)$$

which excludes the cases considered in Eq. (45). The conditions of independence given by Eqs. (42)–(45) can be achieved by minimizing the cost function

$$E=\alpha\sum_{l<j}C_{ij}^2+\beta\sum_{i<j\leq k}C_{ijk}^2+\gamma\sum_{i<j\leq k\leq l}C_{ijkl}^2+\delta\sum_{i<j}(C_{iijj}-C_{ii}C_{jj})^2 \quad (47)$$

where α, β, γ, δ are the inverses of the number of elements in each summation, respectively. It is very easy to test whether a factorized probability distribution [Eq. (7)] satisfies the Eqs. (42)–(45).

B. Minimization of mutual information

The mutual information between the output components is defined as

$$M=\sum_{i=1}^{d}H(x_1,\dots,x_{i-1};x_i)=\sum_j H(y_j)-H(\vec{y}) . \quad (48)$$

Due to the fact that M is a measure of the amount of information between the components of the outputs, it is also a measure of statistical correlations between the components of the outputs. In fact, statistical independence as expressed in Eq. (7) is then equivalent to [18]

$$M=\sum_j H(y_j)-H(\vec{y})=0 . \quad (49)$$

This means that in order to minimize the redundancy at the output we minimize the mutual information between the different components of the output vector. Due to the fact that the herein defined structure of the neural network conserves the entropy, i.e., $H(\vec{y})=H(\vec{x})=\text{const}$, the minimization of M reduces to the minimization of $\sum_j H(y_j)$. The second theorem of Gibbs assures that the entropy of a distribution has an upper bound given by the entropy of a Gaussian distribution with the same variance as the original one. Using this theorem we can reduce the problem of statistical decorrelation to the problem of minimizing the upper bound of the entropies $\sum_j H(y_j)$, i.e., the sum of the output variances (which is the entropy of a sum of Gaussian distributions). If the variance of each component is denoted by $\aleph_j$, then

$$\text{minimization}(M)\equiv\text{minimization}\{\text{Tr}[\ln(\text{diag}(\aleph_j)]\} . \quad (50)$$

In other words, the cost function is defined in this case as

$$E=\sum_j\ln(\aleph_j) . \quad (51)$$

In the particular case that only second-order terms are used in the cumulant expansion, the learning rule reduces to Eq. (51). This expresses nothing else than the diagonalization of the second-order covariance matrix. In this case, by transforming the cumulant expansion of the Fourier transform of the distribution back,

$$P(y_i)=\int dK_i e^{-i(K_iy_i)}(e^{i\aleph_i^{(1)}K_i-(\aleph_i^{(2)}K_i^2)})=\frac{\exp\left[-\dfrac{(y_i-\aleph_i^{(1)})^2}{2\aleph_i^{(2)}}\right]}{\sqrt{2\pi\aleph_i^{(2)}}} , \quad (52)$$

we obtain a Gaussian distribution. Diagonalization of the covariance matrix decorrelates statistically the components of the output only if we assume a Gaussian distribution of the outputs. In the case of Gaussian distributions, minimization of the sum of the variances at each output leads exactly to statistical decorrelation, but in the non-Gaussian case only an upper bound is minimized.

It is important to remark that the optimal embedding dimension is determined by number of points in the past that are correlated statistically with the present. A strategy to measure statistical correlations is by trying to decorrelate statistically (reducing mutual information or expanding in higher-order cumulants, but not only linear decorrelation, i.e., diagonalization of first-order correlation matrix) and than seeing how far the past is needed to decorrelate the present, or in other words, how many points in the past are necessary to model (i.e., find the statistical correlations) the present. This technique is related to the ones proposed by Fraser and Swinney [32] and the works of Liebert and Schuster [22] and Liebert, Pawelzik, and Schuster [23] which formulate for the detection of optimal embedding the minimization of the mutual information. The learning rule for both decorrelation techniques can be easily expressed by the gradient descent method

$$\vec{\omega}(n+1)=\vec{\omega}(n)-\eta\frac{\partial}{\partial\vec{\omega}}E(n)\ . \tag{53}$$

IV. RESULTS AND SIMULATIONS

In this section we present four examples. The first three apply the theory herein developed for the modeling of chaotic time series. The last experiment deals with the modeling of coupled chemical reactions.

The first experiment concerns the chaotic time series

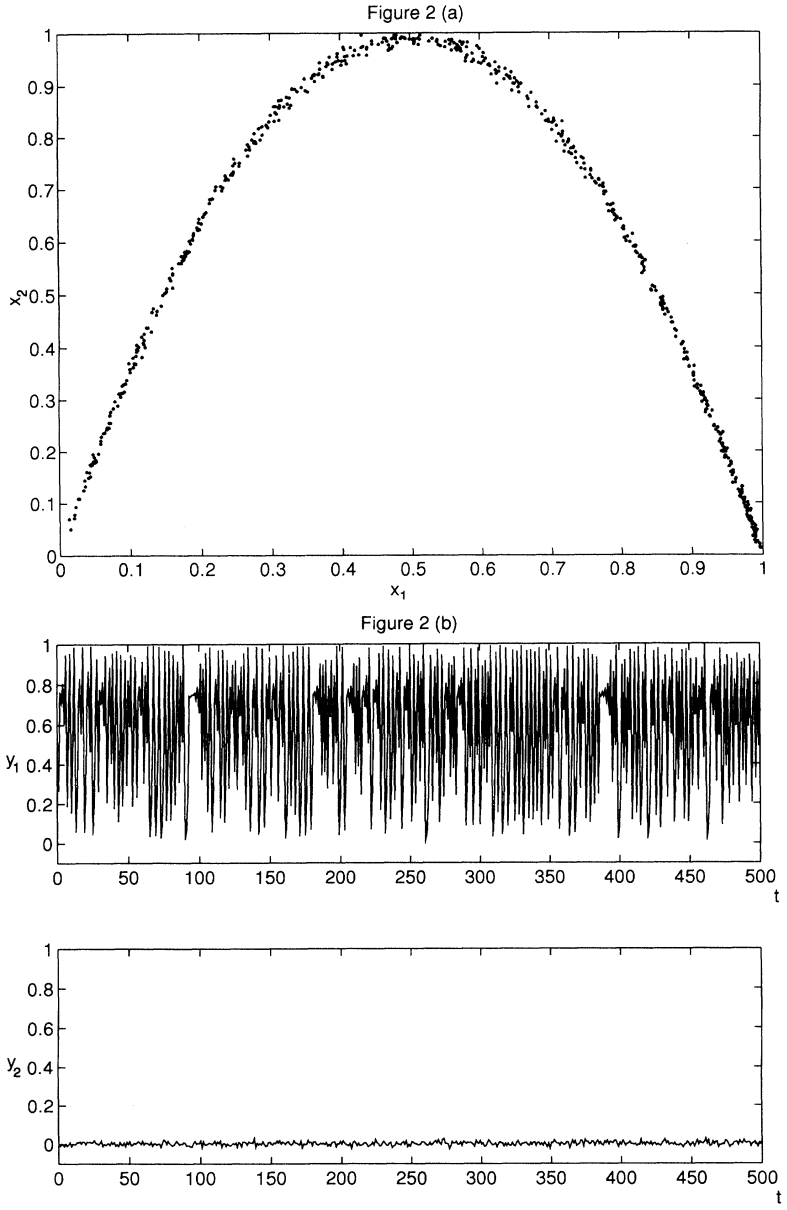

FIG. 2. Input and output space distributions after training with a one-layer polynomial symplectic network of order $R=2$ for the logistic map. (a) Input state space; (b) outputs as a function of time after training two-input and two-output neural network for extraction of correlations between the elements of the embedding vector of the logistic map.

generated by the logistic map. In this case the noisy logistic map was used to generate the input

$$x_2 = 4x_1(1-x_1)+v \tag{54}$$

where v introduces 1% Gaussian noise. It is important to remark that the noise is not white but colored, due to its inclusion in the iterative mapping. In this case a one-layer polynomial network of second order trained with the cumulant expansion techniques was used. The learning constant was $\eta=0.01$, and 10 000 iterations of training were performed. We have used 100 training patterns taken between $t=2000$ and 2500 and generated by Eq. (54). In Fig. 2(a) the input space is plotted. Figure 2(b) shows the evolution in time after training of the output values of the neural transformation. As is clear from Fig. 2(b), the strongly correlated input (past and present of the chaotic time series, i.e., embedding vector with delay equal to two) is decorrelated. All the information was allocated practically in one of the coordinates of the output (y_1), leaving the second (y_2) merely for the representation of the noise (this last coordinate is practically constant in time, meaning that the network extracted the nonlinear correlation between the two inputs, i.e., a nonlinear PCA was performed). Analyzing the weights of the trained network, the polynomial right-hand side (rhs) of Eq. (54) is recovered. This means that the deterministic part that generates the chaotic time series was modeled, even in the presence of colored noise which was decorrelated and allocated in the second coordinate of the output. The second example applies to the Hénon map [33] defined by the following system of iterative equations

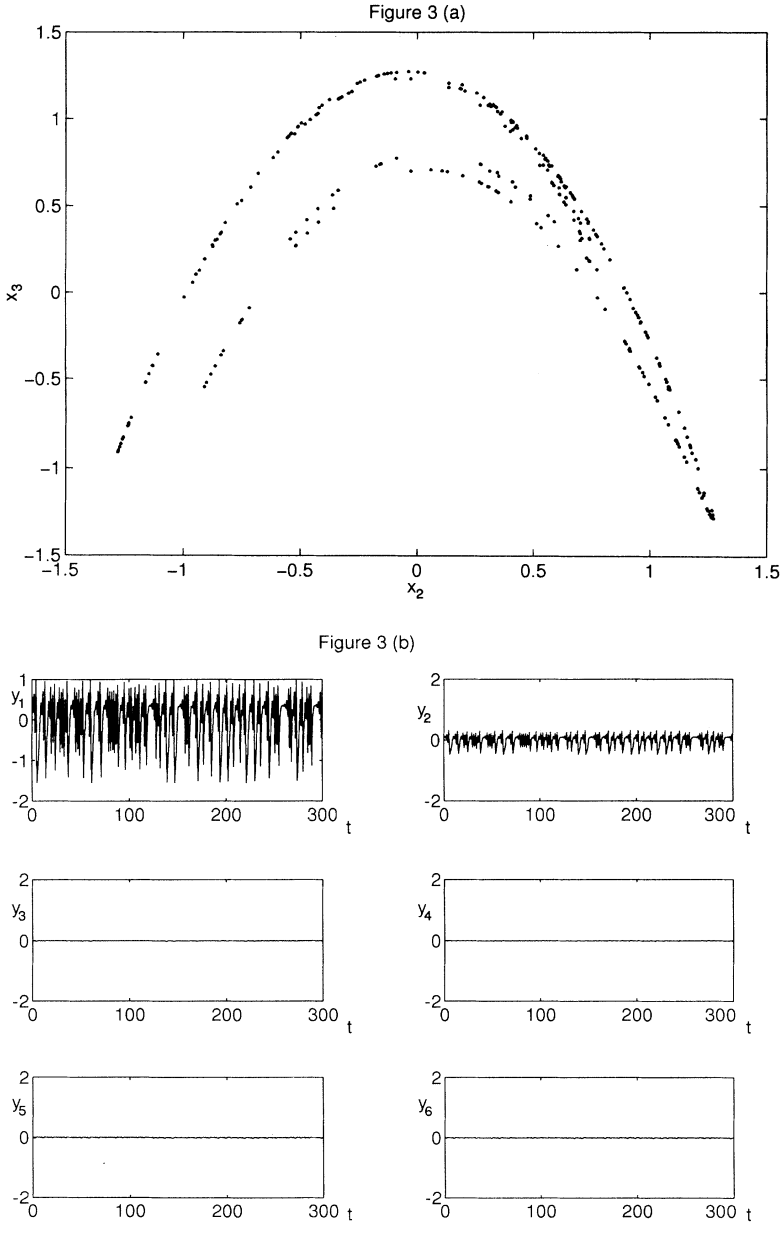

FIG. 3. (a) Poincaré structure of the Hénon map. (b) Outputs as a function of time of a six-input and six-output neural network trained unsupervised for extracting the decorrelation between the component of a six-dimensional embedding vector for the Hénon map. (c) Same as (b) but using standard PCA. (d) Reconstructed Poincaré structure of the Hénon map.

$$x_{n+1}=1-1.4x_n^2+y_n \ , \tag{55}$$

$$y_{n+1}=0.3x_n \ . \tag{56}$$

We analyze the variable x. In this case the exact embedding is known just by inserting (56) in (55)

$$x_{n+1}=1-1.4x_n^2+0.3x_{n-1} \tag{57}$$

which means that the required delay for modeling of the variable x is exactly two points in the past. In this case we take on purpose a six-dimensional embedding vector. The goal of the experiment is to model the Hénon map from experimental (simulated) data and to discover the real required delay. We use in this case a polynomial neural network of second order which was trained with the cumulant expansion techniques. The learning constant was $\eta=0.01$, and 5000 iterations of training were performed. We have used 300 training patterns taken between $t=2000$ and 2500 and generated by integrating Eqs. (55) and (56). Figure 3(a) shows the state-space structure of the attractor by plotting the two successive input signals x_2, x_3. Figure 3(b) shows the evolution in time after training of each output component. It is interesting to note that up to component three the output is constant meaning that the network has figured out how to decorrelate this output using the past. This indicates also that two points in the past are sufficient to model the map. In fact, the polynomial generated by the trained network corresponding to output 3 is the rhs of Eq. (57). The polynomials corresponding to outputs 4, 5, and 6 are also

Figure 3 (c)

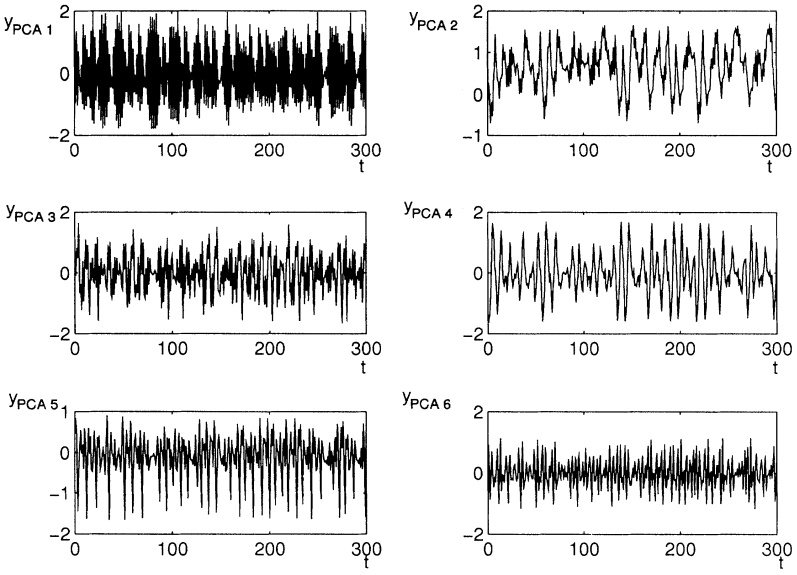

Figure 3 (d)

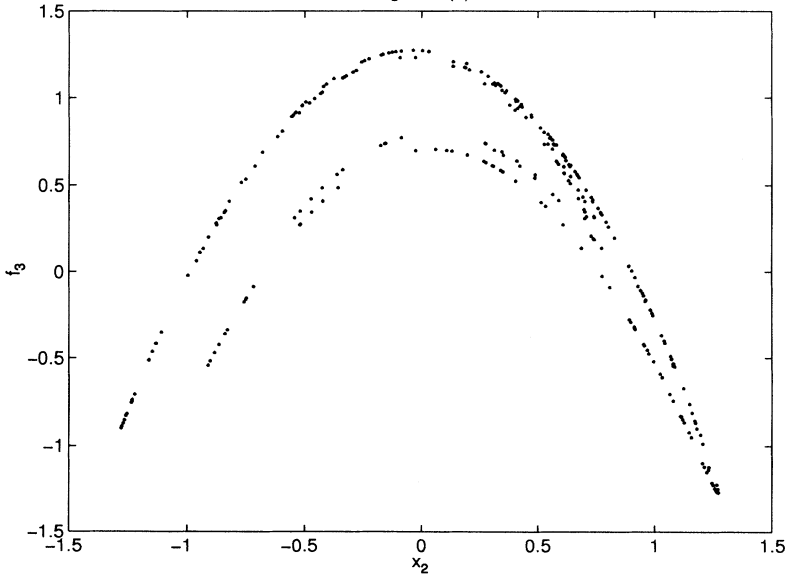

FIG. 3. (*Continued*).

identical to the rhs of Eq. (57), i.e., the terms and inputs too far in the past were automatically pruned by the learning algorithm. Figure 3(d) shows the reconstructed structure of the attractor by plotting the trained function f_3 of Eq. (12) as a function of x_2. The structure plotted in Figs. 3(a) and 3(d) are identical the perfect learning of the structure of the attractor of the dynamical system. In order to compare with the traditional PCA we plotted in Fig. 3(c) of each output component after linear decorrelation (PCA) for the evolution in time after training. As it is clear it is impossible to decorrelate linearly the Hénon map and therefore no constant outputs are observed.

The third example focuses on the same goal as the last example but it is more difficult. The Mackey-Glass system formally has an infinite number of degrees of freedom but its strange attractor has finite dimension. Therefore it mimics a typical experimental situation. The delay difference equation of Mackey-Glass [34] can be expressed as

$$\dot{x}(t) = -bx(t) + \frac{ax(t-T)}{1+x^{10}(t-T)} , \tag{58}$$

where $a=0.2$, $b=0.1$, and $T=30$. We use in this case a polynomial neural network of order 2. The learning constant was $\eta=0.01$, and 25 000 iterations of training were performed. We have used 500 training patterns taken between $t=2000$ and 2500 and generated by integrating Eq. (58). The input and output dimension is six and the technique for learning is also a cumulant expansion. The six inputs are given by $x(t-50)$, $x(t-40)$, $x(t-30)$, $x(t-20)$, $x(t-10)$, $x(t)$. We have chosen the delay time equal to 10 in order to compare our method with the results obtained by Liebert and Schuster [22] and Liebert, Pawelzik, and Schuster [23]. The time delay 10 is obtained by using the principle of maximal statistical decorrelation as is thoroughly explained in Ref. [22]. Let us note that in this case only a quadratic approximation (the first two terms of a Taylor expansion) can be modeled with this network, due to the fact that the neural architecture in this example is a polynomial of second order. The embedding dimension can be detected even with this second order approximation by analyzing the weight connections or output components after training. Those weights which are negligibly small indicate statistical independence. The embedding dimension found is four, in agreement with the results of Liebert and Schuster [22] and Liebert, Pawelzik, and Schuster [23].

Figure 4 shows the evolution in time after training of each output component. Up to component four the output has a very low variance meaning that the second order polynomial network has extracted an approximated form of the correlation of these outputs using the past four points. This indicates also that four points in the past (embedding dimension) are required to approximately model the map by the second order polynomial. The correlation cannot be totally extracted since the original series is nonpolynomial and the network is a second-order polynomial. This is the reason why the components five and six of the output are nonconstant but present still a small oscillation.

It should be remarked that, in general, our method works well only in the case where the nonlinearity that defines the dynamics may be approximated by a polynomial. This implies that in the special case of a polynomial dynamics the underlying equations can be accurately extracted by a polynomial neural network. In other cases *a priori* knowledge about the nonlinearity is necessary to decide on the quality of the polynomial approximation. Alternatively, *a priori* knowledge can be used to construct a volume-conserving triangular architecture that corresponds to the correct form of the nonlinearity.

The last example consists of the use of the symplectic architecture of Fig. 1(b) in order to learn the kinetic constants of a system of coupled chemical reactions. The system used is for the synthesis of Bromide acid and corresponds to the following system of kinetic reactions:

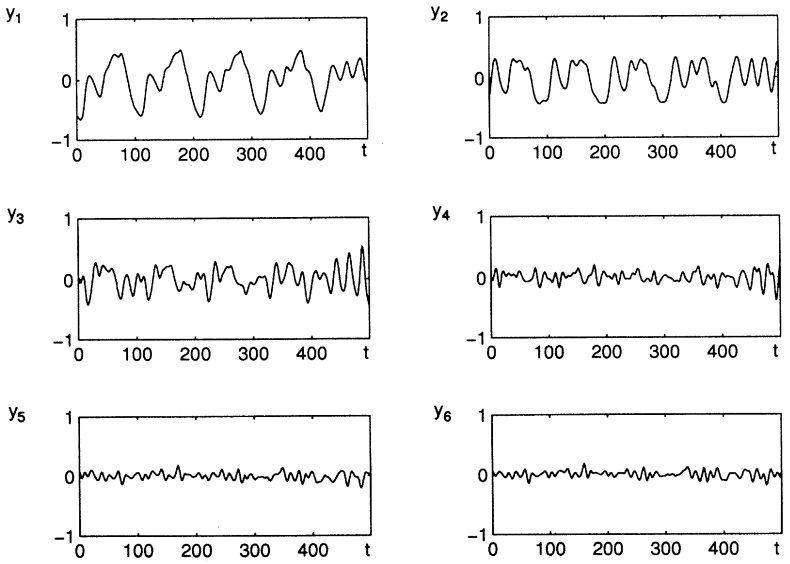

FIG. 4. Outputs as a function of time of a six-input and six-output neural network trained unsupervised for extracting the decorrelation between the component of a six-dimensional embedding vector for the Mackey-Glass time series.

$$\frac{d}{dt}[\mathrm{Br}_2]=k_5[\mathrm{Br}]^2-k_1[\mathrm{Br}_2]\ ,$$
$$\frac{d}{dt}[\mathrm{Br}]=2k_1[\mathrm{Br}_2]-2k_5[\mathrm{Br}]^2+k_9[\mathrm{HBr}]-k_8[\mathrm{H}][\mathrm{Br}]\ ,$$
$$\frac{d}{dt}[\mathrm{H}_2]=k_7[\mathrm{H}]^2-k_6[\mathrm{H}_2]\ , \tag{59}$$
$$\frac{d}{dt}[\mathrm{H}]=2k_6[\mathrm{H}_2]-2k_7[\mathrm{H}]^2+k_9[\mathrm{HBr}]-k_8[\mathrm{H}][\mathrm{Br}]\ ,$$
$$\frac{d}{dt}[\mathrm{HBr}]=k_8[\mathrm{H}][\mathrm{Br}]-k_9[\mathrm{HBr}]$$

associated with the elementary chemical reactions

$$\mathrm{Br}_2 \underset{k_5}{\overset{k_1}{\leftrightarrow}} 2\mathrm{Br}\ ,$$
$$\mathrm{H}_2 \underset{k_7}{\overset{k_6}{\leftrightarrow}} 2\mathrm{H}\ . \tag{60}$$
$$\mathrm{H}+\mathrm{Br} \underset{k_9}{\overset{k_8}{\leftrightarrow}} \mathrm{HBr}$$

where the real kinetic constants are

$$k_1=9.2\times10^{-5}\ ,\quad k_5=4.0\times10^{15}\ ,\quad k_6=9.2\times10^{-5}\ ,$$
$$k_7=4.0\times10^{15}\ ,\quad k_8=1.0\times10^{15}\ ,\quad k_9=1.0\times10^{-5}\ . \tag{61}$$

Equation (61) was integrated by the Gear method due to the high stiffness of this system for the generation of the simulation data. The equation is modeled in the symplectic structure of Fig. 1(b). The first five inputs are the concentrations of time t, and the last five inputs are the flows. We have used in this case the minimization of the variances of the last five outputs (mutual information argument). The training constant was $\eta=0.1$ and the network was trained during 150 000 epochs using 3500 training patterns obtained from the generated simulation data. After training, the following values for the kinetic constants were obtained:

$$k_1=9.19\times10^{-5}\ ,\quad k_5=3.92\times10^{15}\ ,\quad k_6=9.19\times10^{-5}\ ,$$
$$k_7=3.9\times10^{15}\ ,\quad k_8=1.01\times10^{15}\ ,\quad k_9=1.05\times10^{-5}\ , \tag{62}$$

which are in close agreement with the real ones given by Eq. (61). This close agreement means that the rates of change of the concentrations (as a function of the evolved time) obtained from the original and learned equations are extremely similar as is displayed in Fig. 5 (in fact, they are indistinguishable).

V. CONCLUSIONS

In this paper we have focused on the problem of modeling time series by learning the statistical correlations between the past and present elements of the series in an unsupervised fashion. An information-theoretic-based architecture and learning paradigm was introduced. The neural architecture employed possesses only one layer and a triangular structure in order to transform elements of the input by observing only the past and to conserve the volume. This permits us to devise a transformation that guarantees transmission of information without loss. Two alternative methods were used for formulating a learning rule that decorrelates the output components of the network. After decorrelation between the output components, correlation between the elements of the time series has been extracted by analyzing the trained neural architecture. As a consequence, we were able to model chaotic as well as nonchaotic time series.

A critical point in modeling times series is the determination of the dimension of the embedding vector. With the method described in the present work it is possible to detect the embedding dimension by extracting the correlations of remote past and future. We have tested our methods and techniques on various representative examples, including chaotic time series and chemical reactions.

Our work makes extensive use of fundamental concepts

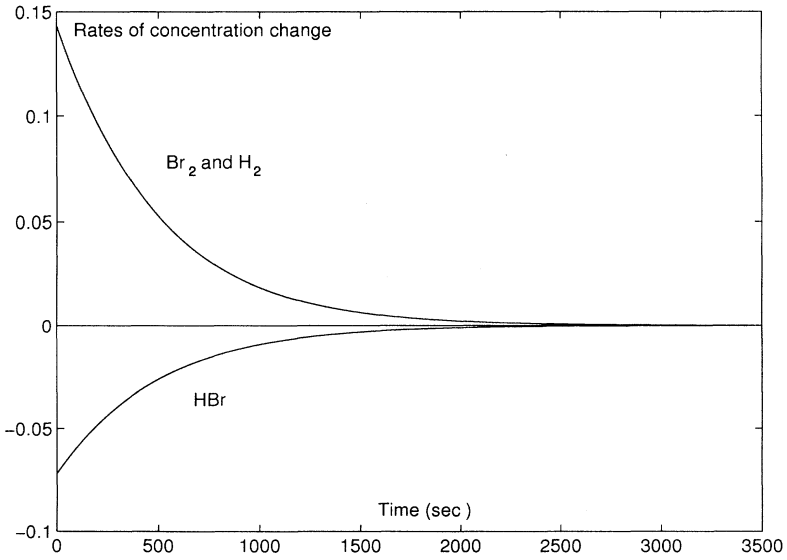

FIG. 5. The rates of change for the concentrations of Br_2, H_2, and HBr as functions of time. The curves obtained from the original and learned equations are indistinguishable.

of information theory, nonlinear dynamics, and neural networks. In particular, it builds upon the seminal work of Fraser and Swinney in that it employs the mutual information concept for the detection of correlations in a time series. It goes beyond other related works by introducing new concepts from the theory of unsupervised learning in neural networks. These concepts enable us to determine at the same time the relevant (embedding) dimension of chaotic dynamics and the equations describing it, and hence they are very powerful. This feature is of particular importance when analytical models of the physical process at hand are not available but experimental data are available (*data driven* modeling). We believe that the methods developed in this work offer a instrument for analyzing the real world data of complex nonlinear processes.

[1] C. Beck and F. Schlögl, *Themodynamics of Chaotic Systems, Cambridge Nonlinear Science Series* (Cambridge University, Cambridge, 1993).

[2] P. Grassberger and I. Procaccia, Phys. Rev. Lett. **50**, 346 (1983).

[3] J. P. Eckmann and D. Ruelle, Rev. Mod. Phys. **57**, 617 (1985).

[4] J. P. Crutchfield and B. McNamara, Complex Systems **1**, 417 (1987).

[5] H. D. I. Abarbanel, R. Brown, and J. B. Kadtke, Phys. Lett. A **138**, 401 (1989).

[6] H. D. I. Abarbanel, R.Brown, and J. B. Kadtke, Phys. Rev. A **41**, 1782 (1990).

[7] J. Farmer and J. Sidorowich, Phys. Rev. Lett. **59**, 845 (1987).

[8] J. Cremers and A. Hübler, Z. Naturforsch. A **42**, 797 (1986).

[9] A. Lapedes and R. Farber, Technical Report No. LA-UR-987-2662, Los Alamos National Laboratory, Los Alamos, NM, 1987 (unpublished).

[10] M. Casdagli, Physica D **35**, 335 (1989).

[11] A. Weigend, D. Rumelhart, and B. Huberman, in *Proceedings of the 1990 Connectionist Models Summer School,* edited by D. Toureztky, J. Elman, T. Sejnowski, and G. Hinton (Morgan Kaufmann, San Francisco, 1990), p. 105.

[12] D. M. Wolpert and R. C. Miall, Philos. Trans. R. Soc. London Ser. B **242**, 82 (1990).

[13] A. Albano, A. Passamante, T. Hediger, and M. E. Farell, Physica D **58**, 1 (1992).

[14] G. Deco and B. Schürmann, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E77-A** (4), 1840 (1994).

[15] H. Barlow, *National Physical Laboratory Symposium No. 10, The Mechanization of Thought Processes* (Her Majesty's Stationery Office, London, 1959).

[16] H. Barlow, Neural Comput. **1**, 295 (1989).

[17] J. Atick and A. Redlich, Neural Comput. **2**, 308 (1990).

[18] J. Atick and A. Redlich, Neural Comput. **4**, 196 (1992).

[19] A. Redlich, Neural Comput. **5**, 289 (1993).

[20] A. Redlich, Neural Comput. **5**, 750 (1993).

[21] T. Buzug and G. Pfister, Phys. Rev. A **45**, 7073 (1992).

[22] W. Liebert and H. G. Schuster, Phys. Lett. A **142**, 107 (1989).

[23] W. Liebert, K. Pawelzik, and H. G. Schuster, Europhys. Lett. **14**, 521 (1991).

[24] K. Pawelzik and H. G. Schuster, Phys. Rev. A **43**, 1808 (1991).

[25] K. Takens, in *Dynamical Systems and Turbulence,* Vol. 898 of *Lecture Notes in Mathematics,* edited by D. A. Rand and L. S. Young, New York (Springer-Verlag, New York, 1980), p. 366.

[26] T. Sauer, J. Yorke, and M. Casdagli, J. Stat. Phys. **65**, 579 (1991).

[27] D. Nychka, S. Ellner, D. McMcaffrey, and R. Gallant, J. R. Statist. Soc. B **54**, 399 (1992).

[28] R. L. Smith, J. R. Statist. Soc. B **54**, 329 (1992).

[29] C. Shannon, Bell System Techn. J. **7**, 379 (1948).

[30] R. Abraham and J. Marsden, *Foundations of Mechanics* (Benjamin-Cummings, London, 1978).

[31] C. W. Gardiner, *Handbook of Stochastic Methods* (Springer-Verlag, Berlin, 1990), 2nd. ed.

[32] A. M. Fraser and H. L. Swinney, Phys. Rev. A **33**, 1134 (1986).

[33] H. Henon, Commun. Math. Phys. **50**, 69 (1976).

[34] M. Mackey and L. Glass, Science **197**, 287 (1977).