Chapter 6
Neural Networks for Signal Processing in Measurement Analysis and Industrial Applications: the Case of Chaotic Signal Processing

Vladimir GOLOVKO, Yury SAVITSKY, Nikolaj MANIAKOV
Laboratory of Artificial Neural Networks, Brest State Technical University
Moskovskaja str. 267, 224017 Brest, Belarus

Abstract. This chapter discusses the use of neural networks for signal processing. In particular, it focuses on one of the most interesting and innovative areas: the chaotic time series processing. This includes time series analysis, identification of chaotic behavior, forecasting, and dynamic reconstruction. An overview of chaotic signal processing both by conventional and neural network methods is given.

6.1. Introduction

Neural techniques have been successfully applied to many problems in the area of signal processing. Different goals and perspectives can be considered in manipulating data sequences generated by physical processes.

Signal filtering is a classical technique to modify the characteristics of the signal itself. Like in traditional approaches, in the neural approaches the signal is observed through a sampling window sliding in time over the signal itself: whenever the observation window photographs a set of signal samples, filtering or transformation is applied and generates the output view of the incoming signal. Many practical examples have been reported in the literature related to various application areas (e.g., in electronics, electrical engineering, mechanical systems, chemical plants, biomedical systems, radio transmissions).

Noise cancellation in a continuous signal is one of the most interesting applications desirable in a wide variety of practical cases. To reduce the noise we can use a finite-duration impulse response (FIR) filter [5], the transformation from the spatial to the frequency space by means of the Discrete Fourier Transform, the Wavelet Shrinkage method [11], or median filtering [12]. However is not always efficient. The use of the ICA (independent component analysis) neural network for extracting noise-free data has been shown a powerful approach [13]. In the case of Gaussian data the PCA (principal component analysis) neural network was shown efficient [14]; it can also be adopted data compression and reduction.

Prediction and cross-correlation abilities of the neural network can be used to reconstruct signals whenever the noise makes the signal poorly understandable or when the sensor observing the signal is occasionally or temporarily not working properly.

Neural implementation of some classical transformations (e.g., Walsh, Hough) has been also studied to exploit the adaptivity of the neural paradigms in configuring the filter coefficients; harmonic signal analysis by neural networks is another high-level transformation.

Signal processing can also be used to extract relevant information from the input signal, e.g., to detect the occurrence of characteristics waveforms, pulses, spikes, and regularities. Several applications are known in speech and sound processing (e.g., automatic typewriters; phoneme and word recognition; speech understanding; automatic translators; voice and sound compression, equalization and manipulation; voice and sound synthesis). Other industrial applications are related to identification of the operating conditions of machinery, plants, and production processes by observing sensor data, and to data cleaning for system diagnosis.

Overviews of different types of neural networks suited for signal processing as well as overviews of their effective applications can be found in [4-14]. Feedforward neural networks are one of the most used, especially the multilayer perceptron (MLP) and the radial basis function networks (RBF) [7-9]. Both of these network types have been shown to be universal function approximators [9] and, consequently, very appropriate for signal processing applications. Another family suited for dynamic modeling is the recurrent neural network (also called time-delay neural network) that was used, e.g., in nonlinear prediction and modeling, adaptive equalization of communication channel, speech processing and measurement [10].

In many real systems (e.g., compound pendula, dripping faucets, predator-prey ecologies, measle epidemics, oscillating chemical reactions, irregular heart beats, stock market, EEG patterns of brainwave activity, central nervous system, physical systems, social behavior), a chaotic behavior has been observed, i.e., a complex, erratic, extremely input-sensitive behavior which cannot be easily understood. Chaos theory is nowadays widely studied and applied in various areas to describe, characterize, and possibly predict the system behavior when such kind of complexity occurs. Due to the increasing interest in these kinds of models and processing, this chapter focuses therefore on chaotic signal processing.

In the system theory, chaotic systems are deterministic models that can be used to describe random, noisy, unpredictable behaviors that are present in natural systems. The behavior of a chaotic system is governed by simple deterministic nonlinear rules that are iteratively applied to generate the next state from the current state and input values; although these rules do not contain any noise, randomness, or probabilities, their repeated application leads to very complex system behaviors in the long term, that cannot be captured by simple global rules. In this sense, unpredictability "emerges" over time.

The chaotic behavior of a dynamical system can be described either by nonlinear mathematical equations or by experimental data. Unfortunately, often we do not know the nonlinear equations that describe the dynamical system. In general we have only experimental signals from the unknown dynamical system. The problem consists therefore in identifying the chaotic behavior and building a model that captures the important properties of the unknown system by using only the experimental data. In order to determine the main properties of our model, we can use the dynamic invariants (namely: correlation dimension, Lyapunov’s exponents, and Kolmogorov’s entropy).

A chaotic system has a sensitive dependence on the initial conditions: starting from very close initial conditions a chaotic system may very rapidly moves to different final states.

Another problem concerning chaotic signals is that they are unpredictable on the long term, because an error at the beginning of prediction increases exponentially in time [1,2]. An improvement of the prediction accuracy is therefore fundamental. Besides, this allows

also for understanding the observed behavior of a nonlinear system and for reconstructing the space of the system states by taking into account the numerical data measured in the system. This is based on the embedding theorem [3], which guarantees that the full knowledge about the system behavior is contained in the time series of characteristic quantities measured in the system; the complete multivariate phase space can be constructed from these time series. The embedding theorem is characterized by some parameters, namely the embedding dimension and the time delay. The estimation of these parameters provides a maximum predictability of the chaotic time series and can be used to choose the optimal window size (number of input samples) to perform forecasting. Neural approaches have been shown effective in forecasting for chaotic systems: examples and techniques will be described in this chapter.

A further problem concerns the chaotic time series processing by using only observed data. From small data samples it is in fact very difficult to reconstruct the system dynamics and to compute the Lyapunov's spectrum. Also in this case neural networks have been shown more powerful for chaotic time series processing than traditional approaches.

To tackle chaotic signal processing by means of neural paradigms, multilayer perceptron (MLP) and radial basis function networks (RBF) [7-9] as well as time-delay neural networks (TDNN) [10] can be applied both to chaotic signal identification and forecasting. The ICA (independent component analysis) and the PCA (principal component analysis) neural networks are suited for advanced filtering [13,14].

Processing of a chaotic signal can be divided in four stages, as shown in Fig. 1. In the first stage the time series analysis is performed to extract the characteristics of the signal. Then the embedding parameters are evaluated to identify the chaotic behavior. Prediction can be performed on the identified model. Finally the phase space reconstruction can take place or the neural network can be built for optimal forecasting.

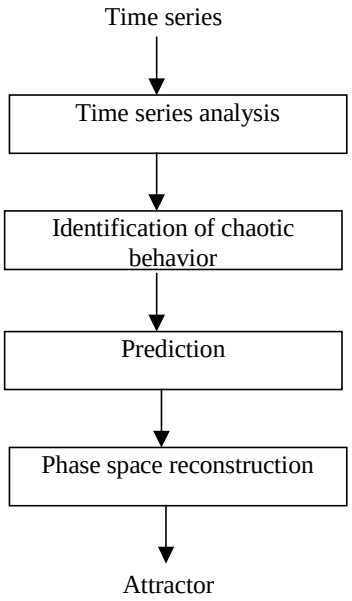


Figure 1: Functional diagram of data processing.

The rest of the chapter is organized as follows. Section 2 reviews the use of multilayer neural networks for signal processing. Section 3 discusses the nonlinear dynamical systems suited for chaotic signal processing and the strange attractor, using Lorenz and Henon data. Section 4 presents different approaches for identification of chaotic behavior. Section 5 describes the time series analysis, namely the computation of the embedding parameters. Section 6 tackles the analytical approaches for computing the Lyapunov's exponents that characterizes the system

chaoticity. Section 7 presents the neural network approach to determine the Lyapunov's spectrum, having very low computational complexity and requiring small data sets. Section 8 introduces the neural network approach for chaotic time series forecasting for individual data points. Section 9 discusses the use of neural network for state space reconstruction.

6.2. Multilayer neural networks

In this section the multilayer neural networks are briefly reviewed with respect to their use in signal processing. Namely, multilayer perceptron, radial basis functions, and recurrent neural networks are discussed.

The multilayer perceptron MLP with one hidden layer is given by:

$$y(l) = \sum_{j=1}^{m} \left(v_{jl} F\left(\sum_{i=1}^{n} w_{ij} x_i - T_j \right) - T_l \right) \quad \text{for } l = \{1, p\} \tag{1}$$

where F is the nonlinear function (sigmoid or other nonlinear), m is the number of hidden units, n and p are the numbers of input and output units respectively, v_{jl} and w_{ij} are the weights, T_j are the thresholds.

The radial basis function network RBF is described by:

$$y(l) = \sum_{j=1}^{m} \left(v_{jl} F\left(\sum_{i=1}^{n} (w_{ij} - x_i)^2 \right) - T_l \right) \quad \text{for } l = \{1, p\} \tag{2}$$

where F is the Gaussian function.

These types of neural networks can be viewed as extensions of the classical approaches to model time series, e.g., the linear autoregressive (AR) model. The AR model is:

$$x(t) = \sum_{i=1}^{n} w_i x(t-i) + \varepsilon(t) \tag{3}$$

where w_i are the weights, $\varepsilon(t)$ is the noise term.

The AR model is equivalent to a linear adaptive filter, based on the least-mean square (LMS). This filter can be built by using a single neuron and the LMS algorithm for its training [4]. The use of an AR model for a stationary processes is based on the Wold's decomposition theorem, according which any discrete-time stochastic process can be decomposed into the sum of a generalized linear process and a predictable process, being each of these processes uncorrelated to the other [5]. The AR model was used in a wide range of signal processing applications, encompassing speech, audio, and images.

Unfortunately, in many real-life applications the signal is generated by nonlinear dynamics and may be non-stationary. Such signals are better represented by nonlinear autoregressive (NAR) processes. These models can be defined by:

$$x(t) = F(x(t-1), x(t-2), ..., x(t-n)) + \varepsilon(t) \tag{4}$$

MLP and RBF networks can be effectively used to model NAR processes. These networks can be configured by appropriate algorithms, e.g., backpropagation and its advanced variations, conjugant gradients, and Levenberg-Marquard.

The recurrent neural networks (RNN), also called time-delay neural networks (TDNN), are extensions of the feed-forward neural networks, obtained by introducing time delays on connections [10] as shown in Fig. 2. This approach was widely used in speech recognition, nonlinear prediction, adaptive equalization of communication channels, and plant control. According to the type of feedback loop, we distinguish the Jordan's network, the Elman's network, and the multi-recurrent neural network.

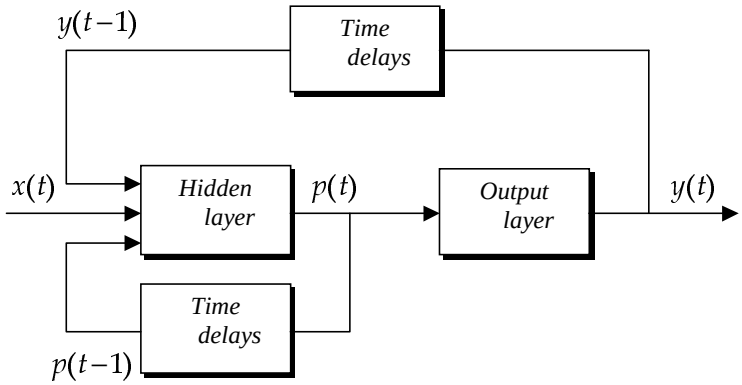


Figure 2: Recurrent neural network

The Jordan's network consists of a multilayer perceptron with one hidden layer and a feedback loop from the output to additional inputs (or context). It computes a nonlinear function of n past sequence elements and q past estimates:

$$y(t) = F(x(t-1), x(t-2), ..., x(t-n), \bar{x}(t-1), \bar{x}(t-2), ..., \bar{x}(t-q)) \tag{5}$$

where $\bar{x}(t)$ is the actual value, and $x(t)$ is the desired value. This model is the nonlinear extension of the ARMA model, i.e., the combination of AR and MA (moving average) components.

The Elman's network has a feedback loop from the hidden layer to additional inputs (or states). This model is described by:

$$\begin{aligned} P(t) &= F(W_1 X(t) + W_2 P(t-1)), \\ Y(t) &= F(W_3 P(t)), \end{aligned} \tag{6}$$

where the matrixes W_1, W_2, W_3 represent three sets of weights: from the input layer to the hidden one, from the hidden layer to the state inputs, and from the hidden layer to the output one.

The multi-recurrent neural network has a feedback both from the hidden and the output layers to the input one. This model is represented by:

$$\begin{aligned} P(t) &= F(W_1 X(t) + W_2 P(t-1) + W_4 Y(t-1)), \\ Y(t) &= F(W_3 P(t)), \end{aligned} \tag{7}$$

where matrix W_4 is set of weights from output layer to context inputs.

The recurrent neural networks may be trained by using of backpropagation algorithm and its more efficient extensions. Another very powerful approach is based on the extended Kalman filter (EKF) [11].

6.3. Dynamical systems

A dynamical system is any object or process for which it is possible to unambiguously define the state at a given time (as the set of characteristic parameters that uniquely identify the system behavior) and the transition function (as the rule that uniquely describes the changes –evolution– of the state in time). The transition function (also called evolution rule) allows for forecasting future states of the dynamical system, starting from a known initial state. A dynamical system may be a mechanical object, a physical system, a chemical substance, a biological entity, a computational process, or any information processing that

is performed according with a deterministic algorithm. Description of the evolution rules can be realized in different way, e.g., by differential equations, discrete mapping, graph theory, or Markov series. The selection of the description method may be lead by to the actual mathematical model of the considered dynamical system.

The *n*-order system can be represent by a set of *n* ordinary differential equations:

$$\frac{d\bar{x}(t)}{dt} = \Phi(\bar{x}(t), t) \tag{8}$$

where $\bar{x}(t) = [x_1(t), x_2(t), ..., x_n(t)]$ is the vector of the system state and Φ is the vector field. The system is called autonomous if the function Φ is not changing in time, i.e.,

$$\frac{d\bar{x}(t)}{dt} = \Phi(\bar{x}(t)) \tag{9}$$

The system state at any time is defined as a point in the *n* dimensional space. The vector field maps a manifold to a tangent space. The integral curve (or trajectory) identifies a flow on the manifold. The set of these flow curves are called orbits.

In the case of linear differential equations the trajectories either asymptotically descend in a phase space to the fixed point or are closed orbits when $t \to \infty$. In the case of a nonlinear function Φ under suitable conditions, the behavior is chaotic and the orbits become the complex subset called strange attractor. Some strange attractors have a known mathematical description (e.g., Lorenz's and Rossler's attractors, Mackey-Glass chaotic time series). Other attractors have been experimentally confirmed to be chaotic but there is no known analytical description (e.g., fluid turbulence, Gravity waves, EEG data).

As an example, the attractor of Lorenz system is shown in Fig. 3; it is described by the following three coupled nonlinear differential equations:

$$\begin{cases} \dfrac{dx}{dt} = G(y - x) \\ \dfrac{dy}{dt} = -xz + rx - y \\ \dfrac{dz}{dt} = xy - bz \end{cases} \tag{10}$$

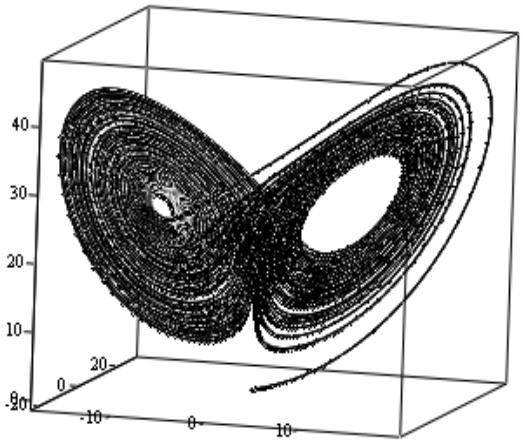
where G=10, r=28, and b=8/3. Lorenz proposed this model for the atmospheric turbulence. An attractor is a subset of the manifold to which an open subset of points (the basin of the attractor) tends to the limit when $t \to \infty$. These are called dissipative systems.

The chaotic flow has a very sensitive dependence on the initial conditions, i.e., points that are initially close each others may exponentially diverge in time. In Fig. 4 two series of a Lorenz system are shown: Series I starts from the initial point [*0, 0.1, 0*], while Series II starts from [*0.001, 0.1, 0*]. A little change in the initial condition leads shortly to different behaviors. This high sensitivity results in unpredictability of the chaotic systems in the long term, since any little inaccuracy is later increased exponentially in time. However, it is important to point out that both of the above series describe the same attractor.

When a dynamical system is described by a first-order differential equation, a chaotic behavior is observed only if the dimension of phase space is greater than 2. However, when the dynamical system is described by difference equation, then the chaotic behavior occurs also in the 2-dimensional space. For the Henon's map:

$$\begin{cases} x_{n+1} = 1 - \alpha x_n^2 + y_n \\ y_{n+1} = \beta x_n \end{cases} \tag{11}$$

where $\alpha = 1.4$ and $\beta = 0.3$, the strange attractor is shown in Fig. 5.



(X, Y, Z)

Figure 3: Lorenz's attractor

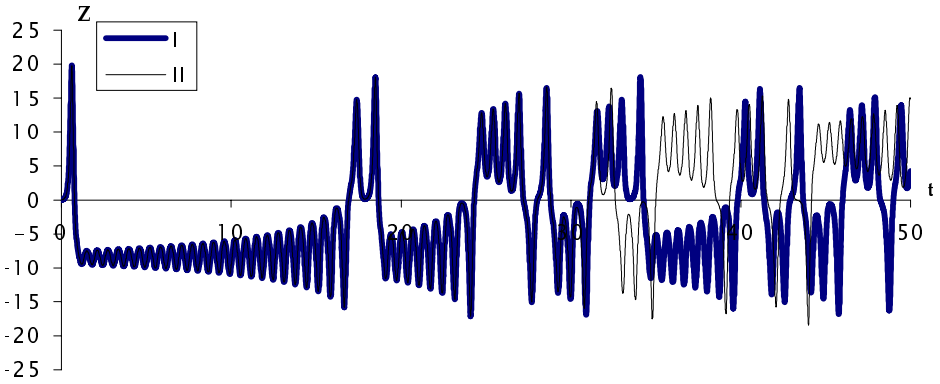


Figure 4: Sensitive dependence on the initial conditions in Lorenz's X-series

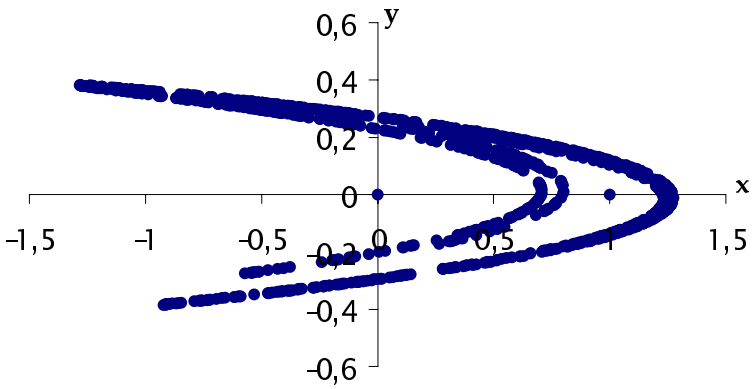


Figure 5: Henon's attractor

As already said, the chaotic behavior can be described either by nonlinear mathematical equations or by experimental data. However, in general only experimental samples of signals produced by the unknown dynamical system are available. The problem consists therefore of identifying the chaotic behavior from these samples and building a model that captures the underlying dynamics of the unknown system. As dynamic invariants correlation dimension, Lyapunov's exponents, and Kolmogorov's entropy can be used.

6.4. How can we verify if the behavior is chaotic?

Chaotic systems include a class of signals that lie between predictable periodic or quasiperiodic signals and totally irregular stochastic signals, which are completely unpredictable. Chaotic processes are deterministic, although they are highly sensible to the initial conditions. Therefore, basic questions in time series analysis are the following: is the process chaotic? How can we distinguish a chaotic process from random (stochastic) or periodic ones? To analyze time series the following methods can be used:

1. the Poincare's section,
2. the power spectrum,
3. the autocorrelation function,
4. the fractal dimension,
5. the Lyapunov's exponents.

One of the easiest ways to identify the chaotic process is to use the Poincare's section. Let's consider a system attractor in the n dimensional phase space and a suited $(n-1)$ dimensional hyperspace S. The orbits of the dynamical system intersect S in points s_k, where k indicate number of consecutive intersections. The coordinates of these points are transformed into the coordinates system associated to the hyperspace. In such way the mapping $F : s_k \rightarrow s_{k+1}$ is obtained. This mapping is called Poincare's mapping (or stroboscopic mapping). By using this method the n-dimensional dynamical system can be reduced to representation with less dimensions, while preserving the main properties of original system. The surface S should be selected so as to maximize the number of intersections, i.e., to minimize the time intervals between them. On a low dimensional representation it is easier to determine the existence of a chaotic behavior. Unfortunately, this approach is not feasible if the equations are unknown.

The fundamental question is to determine the chaoticity of a one-dimensional signal. For this kind of signal we consider the mapping of the consecutive maximum z_k in the time series and we plot the diagram of z_k versus z_{k+1}. This mapping – inspired by the basic concepts of the Poincare's section – reduces the n-dimensional system into only one dimension. By taking the maximum of a coordinate time series we take a sequence of points in which the temporal derivative of the signal is equal to zero: this identifies a hyperspace. For example, by taking the sequence of the maximum in the Z-series of the Lorenz system (Fig. 6), we plot the hyperbolical paraboloid $bz = xy$. In this way a mapping similar to the stroboscopic mapping is therefore built with respect to the considered surface by taking into account only the sequence of z coordinates in real phase space.

By analyzing the diagram of the maximum sequence some conclusions can be derived about the chaoticity. If there is some regularity the process is deterministic, otherwise the process is random. For example, in the diagram of the consecutive maximum in the Lorenz Z-series shown in Fig. 7 the regularity is clear: the behavior is therefore deterministic.

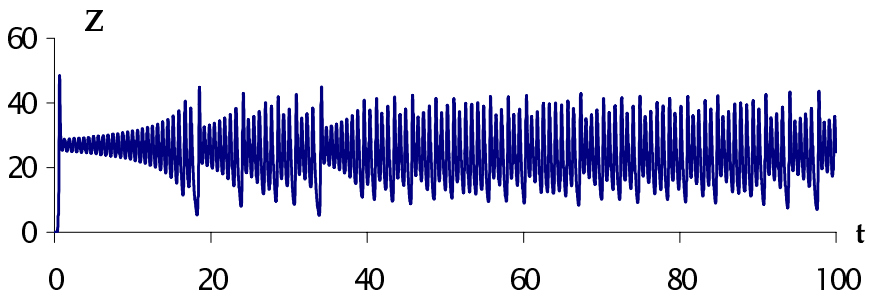


Figure 6: Lorenz's Z-series

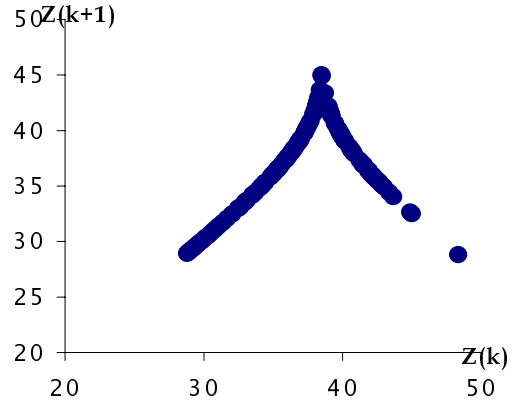


Figure 7: Mapping of the consecutive maximum of the Lorenz's Z-series

Unfortunately, this method is not effective with data affected by noise: for small time derivatives additional extremes can by produced by the perturbations induced by the noise and, consequently, the behavior may become similar to a random process.

Another tools to verify the chaotic behavior is the Fourier transform:

$$x(\varpi) = \lim_{T\to\infty} \int_0^T e^{i\varpi t} x(t)\, dt \tag{12}$$

that transforms the function *x(t)* into the frequency spectrum. The power spectrum is defined as $P(\omega) = \left|x(\varpi)\right|^2$. For a periodic oscillation the power spectrum contains a finite number of frequencies. For a chaotic process it is a broad band. It is worth noting that a multi-harmonic power spectrum does not necessarily correspond to a chaotic system. Systems with a high number of degrees of freedom can generate similar power spectrum.

When a system is represented by samples taken at discrete times in a 2^n period (as in the case of time series), we can use the discrete Fourier transform:

$$x^n(t) = \sum_k a_k^n \cdot e^{2\pi i k t / 2^n} \tag{13}$$

In Fig. 8 the continuous power spectrum of Henon X-series (for 1024 points) is presented: it allows for identifying the chaotic motion or the multi-harmonic oscillation.

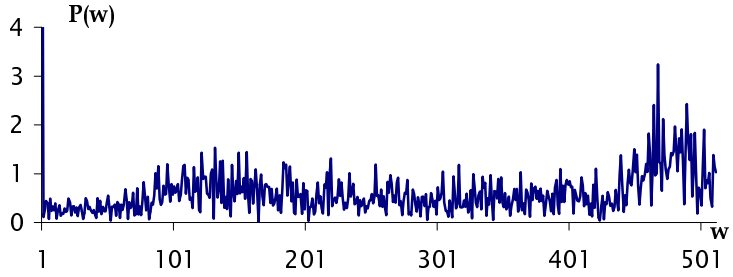


Figure 8: Power spectrum of the Henon's X-series

Another tool to check the process chaoticity is the autocorrelation function, which is defined as follows for continuous and discrete signals, respectively:

$$C(\tau) = \lim_{T\to\infty} \frac{1}{T^2} \int_0^T x(t) \cdot x(t+\tau) dt \tag{14}$$

$$C(\tau) = \lim_{N\to\infty} \frac{1}{N^2} \sum_{t=1}^{N} x(t) \cdot x(t+\tau) \tag{15}$$

In the practice these formulas need to be approximated since only a finite number of point is available. The autocorrelation of a periodical signal produces a periodical function. For chaotic or random signals the autocorrelation function rapidly descend to zero: the autocorrelation function of the Lorenz X-series is shown in Fig. 9.

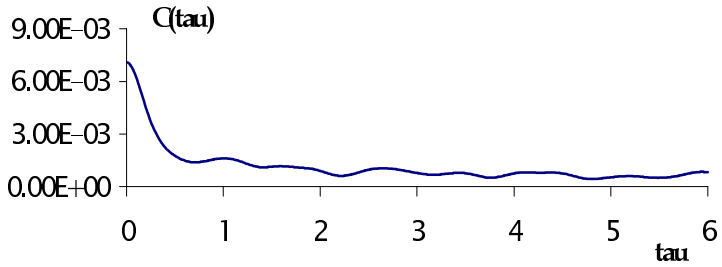


Figure 9: Autocorrelation function of Lorenz's X-series

Another technique to verify the chaotic behavior is the fractal dimension. The attractor of a chaotic process at any time has in fact a fractal dimension. For the one-dimensional observation of the process the correlation dimension D_2 can be evaluated by the algorithm presented in [1]; since this technique requires the use of embedding parameters, it will be discussed in section 5.

The most common and effective test for chaotic behavior verification consists of the Lyapunov's exponent. If the largest Lyapunov's exponent is positive, the process is chaotic; if the sum of all Lyapunov's spectrum is negative the system dissipates and converges to the attractor. The computation both of the largest Lyapunov's exponent and the Lyapunov's spectrum will be discussed in section 6.

6.5. Embedding parameters

A dynamic source of chaotic signals is not fully represented by a one-dimensional observation in the time domain since the chaotic dynamics take place in a phase space having a higher number of dimensions (e.g., for the differential equation it is at least 3). However, the phase space of a chaotic process can be reconstructed from only one time series of observations by using the embedding parameters, as first shown in [2]: the points of time series and their difference (like a derivative) are used as coordinates to build the state space. In [3] the formal proof is given, which is known as the following Time-Delay Embedding Theorem. Let's consider a dynamical system having a solution *(x(t), y(t), …, z(t))* in a *d*-dimensional phase space. By using only one coordinate *x(t)*, under general conditions it is possible to build such space of lag points *(x(t), x(t+τ),x(t+2τ), …, x(t+(D-1)τ))*, that will be a diffeomorphism between itself and the attractor of the dynamical system in the real phase space. The dimension *D* satisfies $D \geq 2\lfloor d_F \rfloor + 1$, where d_F is the fractal dimension of the attractor and $\lfloor . \rfloor$ is the integer part.

The condition $D \geq 2\lfloor d_F \rfloor + 1$ is sufficient but not necessary for reconstructing the dynamics. Besides, also the above theorem assumes that the observable signal is noiseless. However, in the practice signals are noisy time series. Therefore, the experimental data must be preprocessed in order to minimize the influence of noise on the subsequent analysis. To this purpose a FIR filter or an ICA neural network can be used.

The embedding theorem states that even from a single measured signal it is possible to reconstruct the state space that is equivalent to the unknown dynamical system.

To reconstruct the state space the time delay τ and the embedding dimension D must be evaluated. The procedure for finding a suitable D is called embedding. Unfortunately, the embedding theorem does not provide any guidance in choosing the embedding delay τ.

The time delay τ is the period between the components of the points in the reconstructed phase space. The time delay τ should be chosen so that the coordinates of the vectors constituting the embedding space are independent, in order to obtain a faithful reconstruction of the original phase space. In fact if τ is too large, the dynamics at one time step become disconnected from the dynamics at the next time step; consequently, the components of the vector constituting the embedding space will be uncorrelated. The dimension of the reconstructed attractor will be close to the dimension of the embedding space [15] and the attractor will look very complex. This becomes noticeable in the presence of noise: this case is called irrelevance [16]. If τ is too small, all components of the vector will be nearly the same and the attractor will lie close to the line of identity. Consequently, all points will be indistinguishable: this case is called redundancy. All of these cases lead to bad prediction of the chaotic time series.

There are various methods to evaluate the time delay τ:

1. the autocorrelation function,
2. the average displacement method,
3. the mutual information.

The method based on the autocorrelation function $C(\tau)$ is computationally efficient. This approach uses the first zero (or a point which is very close to zero) of the autocorrelation function. The components of the vectors $x(t)$ and $x(t+\tau)$ are thus uncorrelated. Unfortunately, some functions do not reach their first zero in a short time or even do not reach it at all. To avoid this drawback, Zeng advices to take τ as the time at which $C(\tau)$ first falls to e^{-1}/N^2 (where N is the number of points considered in the time series) [17], while Holzfuss suggests to take τ equal to the time at which the autocorrelation function reaches its first minimum [18]. However, these methods do not usually lead to good results because component uncorrelation does not coincide with independency.

The average displacement method [19] estimates the optimum expansion of the reconstructed attractor from the identity line of the reconstructed phase space. To this purpose the following function is used:

$$S(m,\tau) = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\sum_{j=1}^{m-1}\left(x(i\tau + j\tau) - x(i\tau)\right)^2} \tag{16}$$

where N is the number of points in the time series, m is the dimension of the embedding space, and τ is the time delay. For a given m $(m=1,2,...)$, τ is varied until a point in which the function $S(m, \tau)$ reaches a plateau is found. For each dimension of the embedding space a time delay can thus be found.

For higher simplicity, the time delay τ is usually chosen by using the method of mutual information [20], derived from the standard information theory [21]. The set of the time series points is divided into m intervals. The suited number m of intervals is computed by using the Starjes formula: $m \approx \log_2 N + 1 \approx 3.32 \ln N + 1$, where N is the number of points in the time series. The length l of each interval is $l=(x_{max} - x_{min})/m$, where x_{max} and x_{min} are the maximum and the minimum values of the time series, respectively. The mutual information function is defined as:

$$I(\tau) = \sum_{i,j} P_{ij}(\tau) \cdot \ln \frac{P_{ij}(\tau)}{P_i \cdot P_j} \tag{17}$$

where P_i is the probability to observe a value of the time series in the *i*-th interval, and $P_{ij}(\tau)$ is the joint probability that the observed value is located in the *i*-th interval while the subsequent observation after the time τ falls in the *j*-th interval. The function $I(\tau)$ characterizes the probability of observing an $x(t+\tau)$ from the observation of $x(t)$. If the mutual information is equal to zero, no information about $x(t+\tau)$ can be extrapolated. This is equivalent to look for independency of the coordinates' vectors $x(t)$ and $x(t+\tau)$. Unfortunately, it is not possible to find a point in which the mutual information function becomes zero: consequently, the time delay is taken equal to the first minimum of this function. The first minimum of the mutual information function of the Lorenz X-series is 0.16 (Fig. 10).

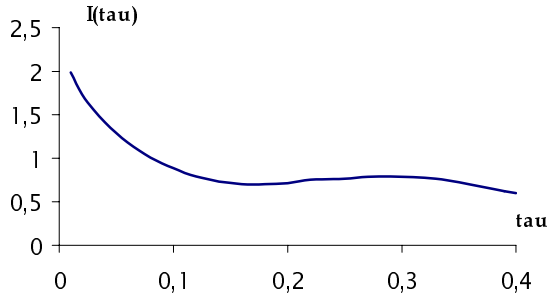


Figure 10: Mutual information $I(\tau)$ versus τ for the Lorenz's X-series

The second fundamental embedding parameter is the dimension *D* of the reconstructed phase space. Various methods can be used for its evaluation:

1. the singular value decomposition,
2. the Takens' theorem,
3. the false nearest-neighbors method.

The singular value decomposition (or principal component analysis, or Karhunen-Loeve decomposition) is effective for a linear system [22]. For this technique, a large embedding space must be considered. The most relevant eigenvalues of the covariance matrix of the embedding samples are taken into account since they characterize the behavior of the system; the others are discarded. The number of these principal components estimates the embedding dimension. Unfortunately, it does not always work for non-linear systems.

The Takens' theorem states that the attractor can be reconstructed from a one-dimensional observation in a phase space with dimension $D \geq 2\lfloor d_F \rfloor + 1$, where d_F is the fractal dimension of the attractor and $\lfloor . \rfloor$ is the integer part. The computation of the dimension *D* consists therefore of computing the fractal dimension d_F. The algorithm for computing the correlation dimension D_2 presented in [1] can be used to this purpose.

The correlation dimension D_2 is an invariant measure defined by:

$$D_2 = \lim_{r \to 0} \frac{\ln Cor(r)}{\ln r} \tag{18}$$

where $Cor(r)$ is the probability that a distance shorter than *r* separates a pair of randomly chosen points [23]. For a point $\bar{x}_1, \bar{x}_2 \ldots \bar{x}_n$ in the phase space, $Cor(r)$ is approximated by:

$$Cor(n,r) = \frac{1}{n(n-1)} \times \left\{ number\ of\ \ pair\ i \neq j\ such, that \left\| \bar{x}_i - \bar{x}_j \right\| < r \right\} =$$

$$= \frac{1}{n(n-1)} \sum_{i,j=1}^{n} H\left(r - \left\| \bar{x}_i - \bar{x}_j \right\|\right) \tag{19}$$

where H is the Heaviside function

$$H(x)=\begin{cases}0, & x<0\\ 1, & x\geq 0\end{cases} \tag{20}$$

For $n\to\infty$, $Cor(n,r)\to Cor(r)$.

To estimate D_2, the diagram of ln(*Cor(n,r)*) versus ln(r) is plotted for the embedding attractor in the phase space having dimension *n=1,2,…*; for each value n, the slope of the curve when the diagram is approximately linear is evaluated. Let's consider increasing values n of the embedding dimension: the value at which the slope becomes constant is the dimension of the one-dimensional observation. Fig. 11 shows log-log diagram of *Cor(n,r)* vs. r for the Henon's X-series for n=1..4. Starting from *n=2* the slope of the straight line is about 1.21; this estimates the correlation dimension for the Henon's attractor. For the Takens' theorem the minimum embedding dimension is $D=2\lfloor d_F \rfloor+1=2\lfloor 1.21\rfloor+1=3$.

For a random process the slope always increases for increasing values of the embedding dimension. This allows distinguish a chaotic process from a random one.

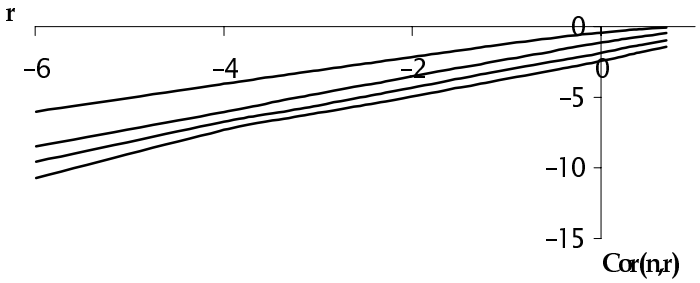


Figure 11: Log-log diagram of *Cor(n,r)* vs. r for the Henon's X-series

The most popular method for estimating the embedding dimension is the False Nearest Neighbors method [24]: the basic idea is related to the non self-intersection of the reconstructed attractor. The original attractor in fact lies on a smooth manifold. The self-intersection of the reconstructed attractor proves that it does not lie on a smooth manifold and, thus, the reconstruction was not correct. For the principle of non self-intersection, when the attractor is reconstructed successfully in R^m, then all neighboring points in R^m should be also the neighbors in R^{m+1}. The method verifies the neighbors for successively higher values of the embedding dimension, until only a negligible number of false neighbors is found when the dimension is increased from m to *(m+1)*. Such m is chosen as the smallest value of the embedding dimension that produces a reconstruction without self-intersections [25].

Formally, for each point $\bar{x}(t)=[x(t),x(t+\tau),...,x(t+(m-1)\tau)]$ of the time series the nearest neighbor $\bar{x}(t_n)=[x(t_n),x(t_n+\tau),...,x(t_n+(m-1)\tau)]$ is identified in the reconstructed phase space of dimension m by using the Euclidean metric:

$$R_m(t,\tau)=\left\|\bar{x}(t)-\bar{x}(t_n)\right\|_m=\sqrt{(x(t)-x(t_n))^2+...+(x(t+(m-1)\tau)-x(t_n+(m-1)))^2} \tag{21}$$

By considering the dimension *(m+1)*, the distance between these points $R_{m+1}(t,\tau)$ is computed. Then, it is:

$$F_t=\sqrt{\frac{R_{m+1}^2(t,\tau)-R_m^2(t,\tau)}{R_m^2(t,\tau)}}=\frac{\left|x(t+m\tau)-x(t_n+m\tau)\right|}{\left\|\bar{x}(t)-\bar{x}(t_n)\right\|_m} \tag{22}$$

If F_t is greater than a given heuristic threshold, this point is marked as false nearest neighbor. By computing the percentage of false nearest neighbors in every dimension m=1,2,…, the dimension D having percentage close to zero is identified. This is the embedding dimension. Fig. 12 shows the diagram of the percentage of false nearest neighbors versus the embedding dimension m for the Lorenz's X-series. In this case the time delay τ is 0.16. From this diagram the minimum embedding dimension for the Lorenz's X-series can be evaluated to be equal to 5, where percentage of false nearest neighbors is 0.3%. A more detailed explanation of this method is given in [26] by using the Gamma test [27].

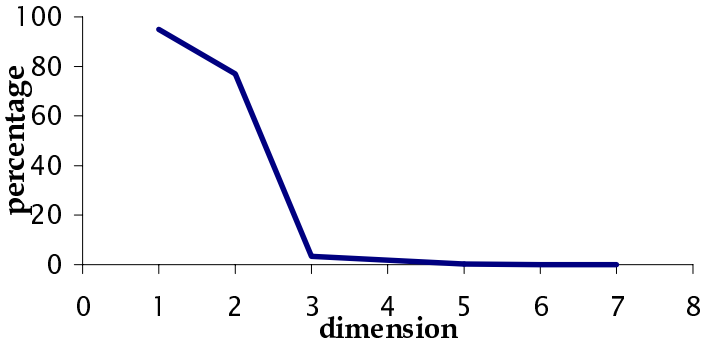


Figure 12: Percentage of false nearest-neighbors points versus embedding dimension m for the Lorenz's X-series

6.6. Lyapunov's exponents

Let's consider a dynamical system described by n differential or difference equations. This system has n Lyapunov's exponents λ_i (i=1,2,…,n), that are globally called Lyapunov's spectrum. The Lyapunov's spectrum describes the system dynamics by defining the evolution of the attractor's trajectories and characterizes the sensitive dependence on the initial conditions. These exponents are the average exponential rates of convergence (divergence) of nearby trajectories in the phase space. The largest Lyapunov's exponent is the statistical measure of the divergence between two orbits starting from slightly different initial conditions. In a chaotic system the largest Lyapunov's exponent is positive.

Let's consider a small sphere at the initial condition in the n-dimensional phase space. Through the time this sphere is transformed into an ellipsoid with n principal axes: the Lyapunov's spectrum measures the exponential growth for the principal axes of the evolving ellipsoid. In fact, let's consider the following Lyapunov's spectrum:

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \tag{23}$$

and let's order the axis of the ellipsoid by decreasing length; λ_1 corresponds to the longest axis, λ_2 corresponds to the subsequent one, and so on. The Lyapunov's exponent λ_i is defined as:

$$\lambda_i = \lim_{t\to\infty} \frac{1}{t} \cdot \ln \frac{l_i(t)}{l_i(0)} \tag{24}$$

where $l_i(0)$ and $l_i(t)$ are the lengths of i-th axis at the initial time and at a time t, respectively. Therefore every Lyapunov's exponent characterizes the modification of the principal axis of the ellipsoid. In an n-dimensional chaotic system the sum of the n Lyapunov's exponents is negative for dissipative systems. The positive exponents are

responsible for the sensitivity to initial conditions. The sum of the positive Lyapunov's exponents is equal to Kolmogorov's entropy.

As already said, the most common test to verify the chaotic behavior consists of checking the highest Lyapunov's exponent: the system is chaotic if such an exponent is positive. When the dynamical system is described by known equations, the highest Lyapunov's exponent can be easily evaluated, e.g., by using the algorithm given in [28].

Let's consider a dynamical system described by the discrete mapping $\bar{x}_{n+1} = F(\bar{x}_n)$, where $\bar{x}$ is the state vector, and n is the index for the discrete time. Starting from an arbitrary point in basin of attraction the mapping is iterated until the obtained point lie on the attractor. This point is $\bar{x}_0$. The nearby point is $\hat{x}_0 = \bar{x}_0 + \tilde{x}_0$, where $\left\|\tilde{x}_0^0\right\| = \varepsilon$ ($\|.\|$ denotes the Euclidean metric). By repeating these operations on the interval T the points $\bar{x}_T$ and $\hat{x}_T$ are derived: the distance vector $\tilde{x}_T = \hat{x}_T - \bar{x}_T$, having length $d_1 = \left\|\tilde{x}_T\right\|$, measures the distance between these two points. d_1/ε characterizes the variations of the perturbation vector at the time T. Then, the point $\bar{x}_T$ is assumed as the new point $\bar{x}_0$ and a new point $\hat{x}_0$ is taken in the direction of the vector $\tilde{x}_T$ so that $\left\|\hat{x}_0 - \bar{x}_0\right\| = \varepsilon$. By repeating these operations the new length d_2 is obtained. After M steps the factor that modifies the amplitude of the perturbation is given by:

$$P = \prod_{k=1}^{M} (d_k/\varepsilon) \tag{25}$$

The highest Lyapunov's exponent can be therefore estimated as:

$$\lambda \cong \frac{1}{MT} \ln P = \frac{1}{MT} \sum_{k=1}^{M} \ln \frac{d_k}{\varepsilon} \tag{26}$$

with M large enough. As a consequence of using a large value for M, the computational complexity becomes high. To limit the computational efforts, the number of iterations should be smaller. To this purpose the variation of the logarithm of the distance d between two nearby point $\bar{x}_0$ and $\hat{x}_0 = \bar{x}_0 + \tilde{x}_0$ is computed in time. By taking in account only the value $d<1$, the straight regression line is identified and its slope can be computed. This estimated slope gives the approximate value of highest Lyapunov's exponent. For example, the highest Lyapunov's exponent for the Henon's system is 0.418, while for the Lorenz's system is 0.906.

Unfortunately, for one-dimensional time series the equations that describe the process are not known. To compute the highest Lyapunov's exponent a different approach must be adopted. First of all, the phase space must be reconstructed from the observation $x(t)$ by using the technique presented above. After having estimated the embedding dimension D and the time delay τ, the lag space $[x(t), x(t+\tau),\ldots, x(t+(D-1)\tau)]$ is built. By taking an arbitrary point in the attractor the nearest point according to the Euclidean metric is identified among the other lag-points. Then the variations of the logarithm of the distance d between these two nearby points are evaluated as discussed above. Finally, the highest Lyapunov's exponent is computed.

The conventional approach to compute λ is as follows:

1. Let's start from two points in the basin of attraction that are separated by the distance d_0. Usually, d_0 is less the 10^{-8}.
2. Execute one iteration for each orbit and compute the new divergence between the corresponding trajectories by using the Euclidean metric. Then evaluate $\ln d_1$.

3. Step 2 is repeated for the n points. $\ln d_2$, $\ln d_3$,…, and $\ln d_n$ are computed.
4. Plot the diagram of $\ln d$ versus n.
5. By using the least square method the straight regression line is drawn, by taking into account only the points having $\ln d < 0$. The slope of the regression line estimates the highest Lyapunov's exponent.

Estimating λ by using this algorithm is –in general– difficult because the initial divergence d_0 is less than 10^{-8}. This approach can be used on experimental data only when the sequence of data is very long: unfortunately, this is usually very difficult to be achieved in real cases. To overcome this limit, the neural networks can be effectively adopted to estimate the highest Lyapunov's exponent.

Another fundamental problem in chaos theory is the computation of the complete Lyapunov's spectra. Its numerical computation can be performed by means of the algorithm presented in [29]. For its estimation the exponential growth of the principal axes of the ellipsoid must be defined.

Let's consider a dynamical system described by n equations. For example, let be n=3. Let's take any point x_0 in the attractor as initial point. Orthonormal frames $\tilde{x}_0^0, \tilde{y}_0^0, \tilde{z}_0^0$ are used for the initial perturbation vectors. After the time T, the trajectory arrives at the point x_1 and the perturbation vector becomes $\tilde{x}_1, \tilde{y}_1, \tilde{z}_1$. The vectors must be reorthonormalized by using the Gram-Schmidt procedure:

$$\begin{aligned}
&\tilde{x}_1^0 = \frac{\tilde{x}_1}{\|\tilde{x}_1\|} \\
&\tilde{y}_1' = \tilde{y}_1 - (\tilde{y}_1 \cdot \tilde{x}_1^0)\tilde{x}_1^0, \quad \tilde{y}_1^0 = \frac{\tilde{y}_1'}{\|\tilde{y}_1'\|} \\
&\tilde{z}_1' = \tilde{z}_1 - (\tilde{z}_1 \cdot \tilde{x}_1^0)\tilde{x}_1^0 - (\tilde{z}_1 \cdot \tilde{y}_1^0)\tilde{y}_1^0, \quad \tilde{z}_1^0 = \frac{\tilde{z}_1'}{\|\tilde{z}_1'\|}
\end{aligned} \tag{27}$$

Then the point x_1 and the perturbation vectors $\tilde{x}_1^0, \tilde{y}_1^0, \tilde{z}_1^0$ are considered. During the next time interval T, the new perturbation vectors $\tilde{x}_2, \tilde{y}_2, \tilde{z}_2$ is obtained. They must be reorthnormalized again. After M steps, the Lyapunov's exponents can be computed as:

$$\lambda_1 = \frac{1}{MT}\sum_{i=1}^{M} \ln\|\tilde{x}_i\|, \quad \lambda_2 = \frac{1}{MT}\sum_{i=1}^{M} \ln\|\tilde{y}_i\|, \quad \lambda_3 = \frac{1}{MT}\sum_{i=1}^{M} \ln\|\tilde{z}_i\| \tag{27}$$

M should be rather large. By using this method the Lyapunov's exponents for the Henon's system are equal to 0.418 and -1.622, while the ones for the Lorenz's system are 0.906, 0 and -14.572.

6.7. A neural network approach to compute the Lyapunov's exponents

The use of neural networks for computing the highest Lyapunov's exponent and the Lyapunov's spectrum was presented in [30]; it relies on the evaluation of the divergence between two orbits at n step ahead by means of an iterative approach.

The neural network for the highest Lyapunov's exponent is a multilayer network with $k \geq D - 1$ input units (where D is the embedding dimension), p hidden units, and one output unit (Fig. 13). This network is trained by means of the sliding window method:

$$x(t+i\tau)=F(x(t+(i-1)\tau), x(t+(i-2)\tau),\ldots, x(t+(i-k)\tau)) \quad \text{for } i = \overline{1,n} \tag{28}$$

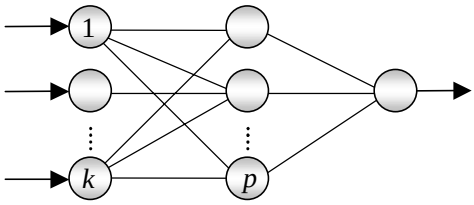


Figure 13: Predicting neural network

Starting from any point of the state space, this neural network finds the nearest –as much as desired– attractor. The highest Lyapunov's exponent by using a small data set can therefore be computed as follows [30]:

1. From the training set a point $[x(t),x(t+\tau),..,x(t+(D-2)\tau)]$ that lies nearby the attractor is chosen and its trajectory $x(t+(D-1)\tau), x(t+D\tau),...$ is computed by using the multistep prediction.
2. In the reconstructed phase space the nearby point $[x(t),x(t+\tau),..,x(t+(D-2)\tau)+d_0]$, where $d_0 \approx 10^{-8}$, is selected and its behavior $x'(t+(D-1)\tau), x'(t+D\tau),...$ is predicted by using the neural network.
3. Define $\ln d_i = \ln\left|x'(t+(D-2+i)\tau) - x(t+(D-2+i)\tau)\right|$, $i=1,2\ldots$, and mark the points for which $\ln d_i < 0$.
4. Plot the diagram $\ln d_i$ versus $i\tau$.
5. Build the regression line for the marked points and compute its slope, which is equal to the highest Lyapunov's exponent.

By using this technique the highest Lyapunov's exponent for the Henon's and the Lorenz's time series are 0.43 and 0.98, respectively. Only the X-series has been used in both cases; the size of the data set was 70 and 100 points, respectively. This result is very close to the actual values computed in the previous section. This method is highly advantageous as computational complexity, accuracy, and small data set are concerned. Figg. 14 and 15 represent the diagram of $\ln d_i$ versus $i\tau$ and the straight regression line for the Henon's and Lorenz's X-series, respectively.

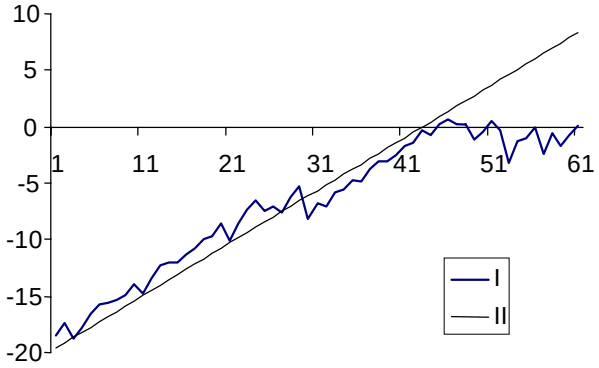


Figure 14: (I) The evolution of the distance between two nearby orbits for the Henon's X-series; (II) the regression line

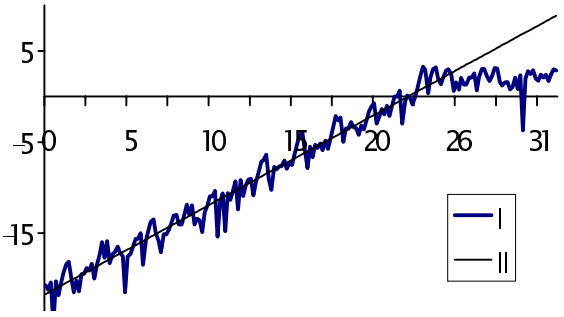


Figure 15: (I) The evolution of the distance between two nearby orbits for the Lorenz's X-series; (II) the regression line

The Lyapunov's spectrum can be computed in a similar way by using an observable time series. Let's consider a dynamical system described by the *n*-dimensional observable vector $x(t)=[x_1(t),x_2(t),\ldots,x_n(t)]$ and assume that the observations $x_i(t)$ are known. A neural network can be created to forecast the next state of dynamical system from the previous one. This network is a multilayer network with *n* input units, *m* hidden units, and *n* output units (Fig. 16). The output is defined as $x(t+1) = F(x(t))$.

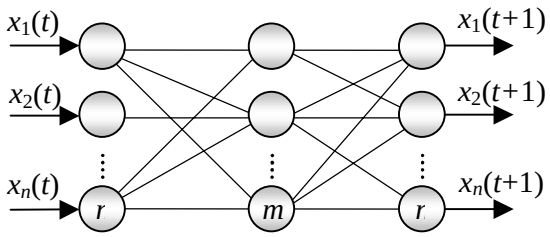


Figure 16: Predicting neural network

Starting from a given initial condition, this network is able to compute the state of the dynamical system at any time, as well as to describe the evolution of the phase trajectory points. At each step the Gram-Schmidt orthogonalization procedure must be used to adjust the output vector. Let $|w_i(t)|$ be the length of the *i*-th vector at the time *t*. This length characterizes the value of the vector along the *i*-th ellipsoid axis. Thus, the *i*-th Lyapunov's exponent is given by:

$$\lambda_i = \lim_{p\to\infty} \frac{1}{p} \sum_{t=1}^{p} \ln \frac{|w_i(t)|}{|w_i(t-1)|} \tag{29}$$

The correspondent length $|w_i(t)|$ can be evaluated by using a neural network and, consequently, the Lyapunov's exponents can be estimated. The algorithm to compute the complete Lyapunov's spectrum is as follows:

1. Take the initial point $N(0)=[x_1(0),x_2(0),\ldots,x_n(0)]$ from the basin of attraction.
2. Choose a small value $\varepsilon \approx 10^{-8}$ and define the coordinates of next *n* points as follows:

$$\begin{aligned} &A_1(0)=[x_1(0)+\varepsilon,x_2(t),\ldots,x_n(t)]\\ &A_2(0)=[x_1(0),x_2(t)+\varepsilon,\ldots,x_n(t)]\\ &\ldots\\ &A_n(0)=[x_1(0),x_2(t),\ldots,x_n(t)+\varepsilon] \end{aligned} \tag{30}$$

   The following orthogonal vectors are obtained:

$$\begin{aligned} &NA_1(0)=[\varepsilon,0,\ldots,0]\\ &NA_2(0)=[0,\varepsilon,\ldots,0]\\ &\ldots\\ &NA_n(0)=[0,0,\ldots,\varepsilon] \end{aligned} \tag{31}$$

3. Compute the length of each vector $|NA_i(0)| = |w_i(0)| = \varepsilon$, where $i = \overline{1,n}$.
4. At the time *t=0*, use the set of points $N(0)$, $A_1(0)$, $A_2(0)$,…, $A_n(0)$ as the input vector of the neural network. The output produced by the predicting network is the set of the coordinates of the points at the next time *t=t+1*:

$$\begin{aligned} &N(1)=[x_1(1,N),x_2(1,N),\ldots,x_n(1,N)]\\ &A_1(1)=[x_1(1,A_1),x_2(1,A_1),\ldots,x_n(1,A_1)]\\ &A_2(1)=[x_1(1,A_2),x_2(1,A_2),\ldots,x_n(1,A_2)]\\ &\ldots\\ &A_n(1)=[x_1(1,A_n),x_2(1,A_n),\ldots,x_n(1,A_n)], \end{aligned} \tag{32}$$

   where $x_j(1,A_j)$ is the *j*-th coordinate of the point A_j at the time *t=1*. This leads to the next set of vectors:

$$\begin{aligned} &NA_1(1) = w_1(1) = [w_{11}, w_{21},\ldots, w_{n1}]\\ &NA_2(1) = w_2(1) = [w_{12}, w_{22},\ldots, w_{n2}]\\ &\ldots\\ &NA_n(1) = w_n(1) = [w_{1n}, w_{2n},\ldots, w_{nn}] \end{aligned} \tag{33}$$

where w_{ij} is the i-th coordinate of the j-th vector, having defined $w_{ij} = x_i(1, A_j) - x_i(1, N)$.

5. The basis $[w_1(1), w_2(1), \ldots, w_n(1)]$ is transformed into the orthonormal frame by using the Gram-Schmidt algorithm, as follows:
   a) The first vector of the orthonormal frame is chosen as:

$$w_1'(1) = \left[\frac{w_{11}}{|w_1(1)|}, \frac{w_{21}}{|w_1(1)|}, \ldots, \frac{w_{n1}}{|w_1(1)|} \right] \tag{34}$$

   where $|w_1(1)| = \sqrt{w_{11}^2 + w_{21}^2 + \ldots + w_{n1}^2}$.

   b) The subsequent vectors are defined by the following recurrent formulas:

$$\begin{aligned} w_i(1) &= w_i(1) - \sum_{j=1}^{i-1} (w_i^T(1) \cdot w_j'(1)) \cdot w_j'(1) \\ |w_i(1)| &= \sqrt{w_{1i}^2 + w_{2i}^2 + \ldots + w_{ni}^2} \\ w_i'(1) &= \left[\frac{w_{1i}}{|w_i(1)|}, \frac{w_{2i}}{|w_i(1)|}, \ldots, \frac{w_{ni}}{|w_i(1)|} \right] \end{aligned} \tag{35}$$

   where $i = \overline{2, n}$.

   c) Compute:

$$s_i(1) = \ln \frac{|w_i(1)|}{|w_i(0)|} \tag{36}$$

   where $i = \overline{1, n}$.

   The result is the new set of points:

$$\begin{aligned} &N(1) = [x_1(1, N), x_2(2, N), \ldots, x_n(1, N)] \\ &A_1(1) = [\bar{x}_1(1, A_1), \bar{x}_2(2, A_1), \ldots, \bar{x}_n(1, A_1)] \\ &A_2(1) = [\bar{x}_1(1, A_2), \bar{x}_2(2, A_2), \ldots, \bar{x}_n(1, A_2)] \\ &\ldots \\ &A_n(1) = [\bar{x}_1(1, A_n), \bar{x}_2(2, A_n), \ldots, \bar{x}_n(1, A_n)], \end{aligned} \tag{37}$$

   where $\bar{x}_j(1, A_j) = w_{ij} + x_i(1, N)$.

6. Repeat from step 3 to step 5 for $t = \overline{1, p}$, where $p \approx 1000$.

7. Define the Lyapunov's spectrum as:

$$\lambda_i = \frac{1}{p} \sum_{t=1}^{p} s_i(t) \tag{38}$$

   where $i = \overline{1, n}$. The following Lyapunov's exponents are therefore obtained:

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \tag{39}$$

By using this approach, the Lyapunov's exponents of the Henon's time series are 0.442 and –1.625 (the actual values are 0.418 and -1.622, respectively). For the Lorenz's time series they are 0.777, 0.003, and –14.472 (the actual values are 0.906, 0, and –14.472, respectively). Figg. 17 and 18 show the dependence of λ_i from p for the Henon's and the Lorenz's time series, respectively.

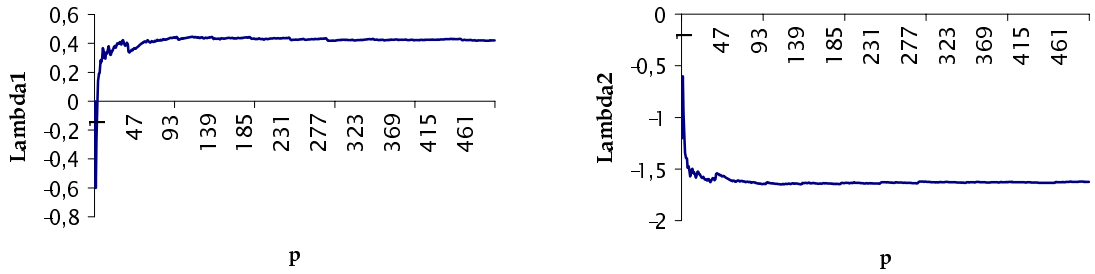


Figure**17:** Estimation of the Lyapunov's spectrum for the Henon's time series

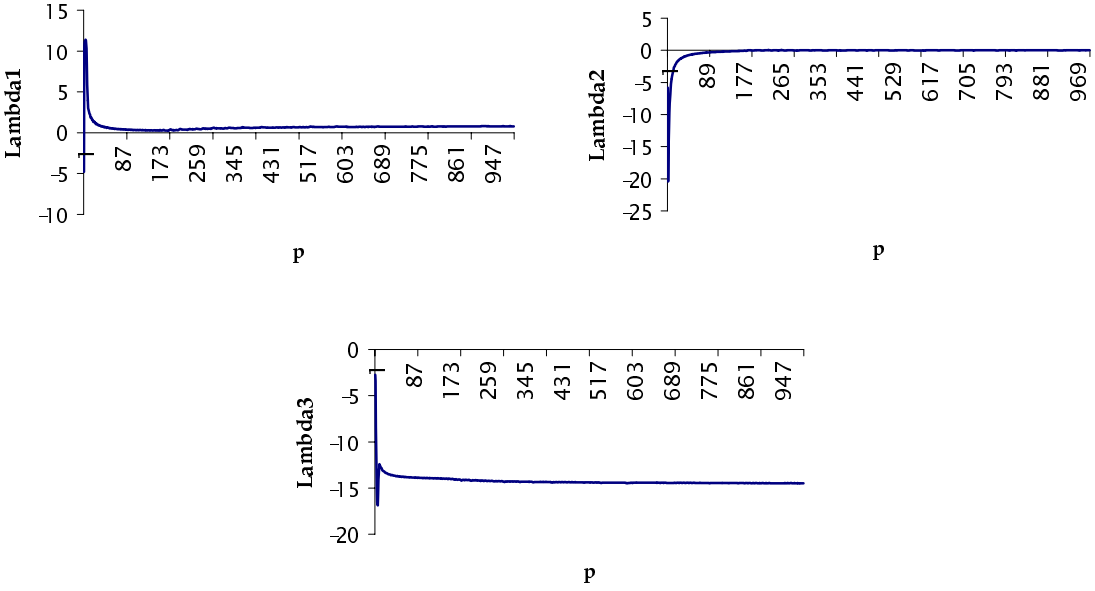


Figure**18:** Estimation of the Lyapunov's spectrum for the Lorenz's time series

6.8. Prediction of chaotic processes by using neural networks

Prediction of a time series consists of finding the sequence *x(l+1), x(l+2)…* that follows a given sequence *x(1), x(2)…, x(l).* The nonlinear predictive model is formally defined by:

$$x(t) = F\left(x(t-1), x(t-2), \ldots, x(t-k)\right) \tag{40}$$

where $t = \overline{k+1, N}$, F is the nonlinear prediction function, and k is the size of the sliding window.

The Multilayer Perceptron (MLP) can be effectively adopted for time series prediction, also for chaotic case. The input layer is composed by at the least *(D-1)* units (where *D* is the embedding dimension), while one output unit delivers the predicted output. The network is trained by using the known data sequence [*x(t), x(t+τ),…, x(t+(D-2)τ)*] to generate the predicted output *x(t+(D-1)τ).* This structure of the predicting network derives directly from the meaning of embedding. When the time series has been learnt by ("embedded in") the neural network, in *D*-dimensional phase space such manifold is obtained that –for every *D-1* coordinates of any point– the *D*-th coordinate is produced and the nearby points in the *D-1* dimensions are very close to the *D*-th coordinate (i.e., the mapping is smooth).

To obtain the maximum predictability the embedding parameters must be defined. Let's consider the Lorenz's and the Henon's attractors as chaotic systems to be modeled. The Lorenz's attractor is defined by the three-coupled differential equations (10); this system is chaotic for G=10, r=28, and b=8/3. Equations (10) can be solved by using a 4-th order Runge-Kutta approach with time step 0.01; Fig. 4 shows the Lorenz's time series (x-axis). The mutual information allows for computing $\tau=0.16$, while the method of the false nearest neighbors evaluates the embedding dimension $D=5$. The window size must be $k \geq D-1=4$. The Henon's attractor is described by the equations (11), where the chaotic behavior occurs for $\alpha=1.4$ and $\beta=0.3$; Fig. 19 shows the Henon's X-series. By using the same reasoning discussed above, the windows size is $k \geq 2$ and $\tau=1$.

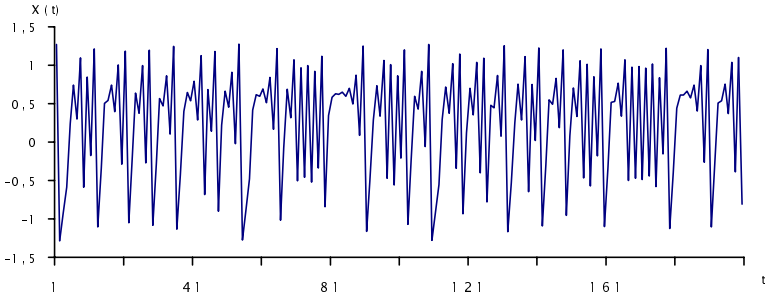


Figure 19: The Henon's X-series (first 200 elements)

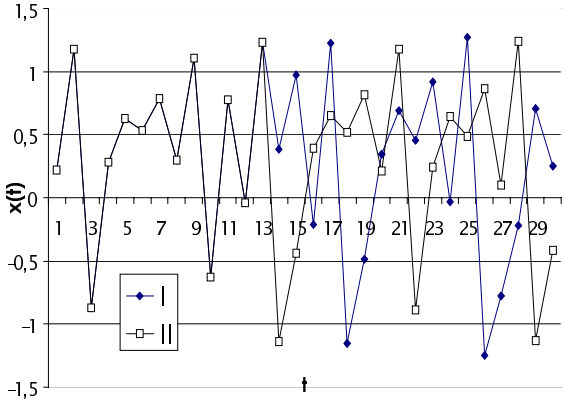


Figure 20: The Henon's process. Prediction results for 30 predicting iterations by using the retraining approach: (I) prediction, (II) original time series

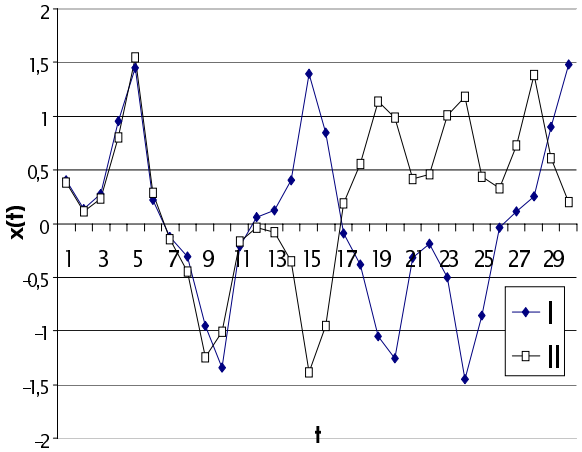


Figure 21: The Lorenz's process. Prediction results for 30 predicting iterations by using retraining approach: (I) prediction, (II) original time series

To perform forecasting at the level of individual points the MLP can be adopted. A neural network with 7 input units, 5 sigmoid hidden units, and 1 linear output unit is verified sufficient to perform this task [30]. Efficient backpropagation is used for training. By using the iterative approach the Henon's and the Lorenz's data series have been predicted for 1500 step ahead; the training set consists of 1500 and 930 patterns for the Henon's and the Lorenz's time series, respectively. Figg. 20 and 21 show the prediction results on 30 steps ahead for the Henon's and the Lorenz's time series, respectively: prediction at the level of the individual data points is unreliable. This unpredictability is one of the main characteristics of a chaotic system.

Prediction can span over a longer time than the individual point. The prediction horizon is the interval of time in which an accurate forecasting is feasible. As said before chaotic data are unpredictable on the long term because the measurement error at the initial condition grows exponentially in time. Since this sensitive dependence is given by a positive value of the highest Lyapunov's exponent, such a value determines the upper prediction limit. It is well known that the sum of all positive Lyapunov's exponents is equal to the Kolmogorov's entropy [31]; consequently, according to the chaos theory, the prediction horizon is [31]:

$$T \approx \frac{1}{K} \cdot \ln\left(\frac{1}{d_0}\right) \tag{41}$$

where $K = \sum_i \lambda_i$ is the Kolmogorov's entropy, $\lambda_i > 0$, and d_0 is the initial prediction error.

According to equation (41), accurate prediction can be achieved only in the range T. Therefore, after having trained the neural network the prediction horizon for the given initial point can be computed. Prediction will be performed with such a horizon to ensure accuracy.

To increase the prediction horizon a suited retraining of the neural network can be performed. Let's assume that the neural network was trained by using the data set $X=\{x(1),x(2),..,x(N)\}$. Prediction will be accurate only for T points ahead: $x(N+1),x(N+2),\ldots,x(N+T)$. The new training set for retraining is $X'=\{x(1),x(2),..,x(N+T)\}$: this allows for extending the prediction horizon. The effectiveness of this approach has been tested for the Henon's and the Lorenz's time series. Tables 1 and 2 show the results achieved with the iterative and the retraining approaches for the Henon's time series. MSE1 and MSE2 are the mean square error for the predicted points $x(N+1),x(N+2)$, $x(N+3)$, $x(N+4)$ and $x(N+5)$, $x(N+6)$, $x(N+7)$, $x(N+8)$, respectively; MSE is the total mean square error and NIT is the number of the training iteration. Tables 3 and 4 show similar results for the Lorenz's time series. The retraining approach usually achieves a better prediction accuracy than the iterative approach and is effectively able to extend the prediction horizon.

6.9. State space reconstruction

Let's finally consider the reconstruction of the state space for a chaotic process by using neural networks, in the presence of a small training data set. Figg. 3 and 5 show the original Lorenz's and Henon's attractors, respectively.

A multilayer perceptron with 7 input units, 5 hidden units, and 1 output unit has been used. The training set consists of 100 and 200 patterns for the Henon's and the Lorenz's time series, respectively. With this neural network, after 3000 training iteration, the mean square errors for the Henon's and the Lorenz's time series are 0.00033 and 0.0008, respectively. Based on the iterative approach the Henon's and the Lorenz's data have been

Table 1: Iterative approach for training the predictive neural network for the Henon's series

Approach	NIT	Size of training set	MSE	T	MSE1	MSE2
Iterative approach	308	950	$3 \cdot 10^{-4}$	4	0.0002227	0.0311980
Retraining approach	276	954	$3 \cdot 10^{-4}$	4	0.0000332	0.0080427

Table 2: Retraining approach for training the predictive neural network for the Henon's series

Approach	Actual value	Desired value	Absolute error
Iterative approach	0.365621	0.363170	0.002451
	0.992627	1.002511	0.009884
	-0.274204	-0.298088	0.023884
	1.191078	1.176354	0.014724
	-1.101723	-1.026758	0.074965
	-0.363043	-0.123019	0.240024
	0.512435	0.670785	0.158350
	0.524174	0.333160	0.191014
Retraining approach	0.364677	0.363170	0.001507
	1.001295	1.002511	0.001216
	-0.288775	-0.298088	0.009313
	1.182933	1.176354	0.006579
	-1.046040	-1.026758	0.019282
	-0.247162	-0.123019	0.124143
	0.592083	0.670785	0.078702
	0.434126	0.333160	0.100966

Table 3: Iterative approach for training the predictive neural network for the Lorenz's series

Approach	NIT	Size of training set	MSE	T	MSE1	MSE2
Iterative approach	1000	800	0.001357	5	0.0053618	0.1628954
Retraining approach	578	805	0.0014	5	0.0011142	0.0698684

Table 4: Retraining approach for training the predictive neural network for the Lorenza's series

Approach	Actual value	Desired value	Absolute error
Iterative approach	-0.155480	-0.163600	0.008120
	-0.556713	-0.617800	0.061087
	-1.573766	-1.633100	0.059334
	-0.536221	-0.439700	0.096521
	0.085535	0.186400	0.100865
	0.237657	0.520500	0.282843
	0.719185	1.254000	0.534815
	1.509935	0.938200	0.571735
	0.461715	0.245600	0.216115
	-0.042810	0.230800	0.273610
Retraining approach	-0.169124	-0.163600	0.005524
	-0.613167	-0.617800	0.004633
	-1.598149	-1.633100	0.034951
	-0.430258	-0.439700	0.009442
	0.121533	0.186400	0.064867
	0.317614	0.520500	0.202886
	0.940051	1.254000	0.313949
	1.336355	0.938200	0.398155
	0.301510	0.245600	0.055910
	0.011798	0.230800	0.219002

predicted for 1500 step ahead. The predicted Lorenz's and Henon's attractors are shown in Figg. 22 and 23: the neural network is able to capture the underlying properties of the chaotic behavior and, therefore, can be used for an accurate reconstruction of the state space and an accurate prediction of the system behavior.

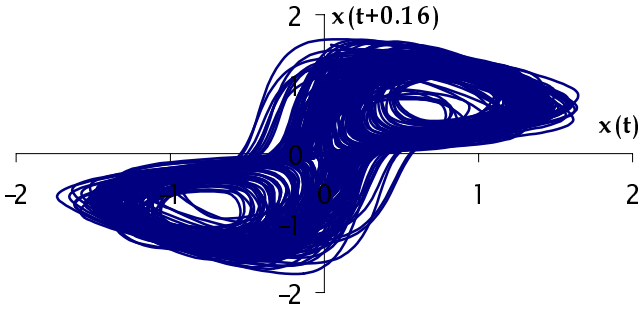


Figure 22: The predicted Henon's attractor: it was built on 1500 predicting iterations in the embedding space

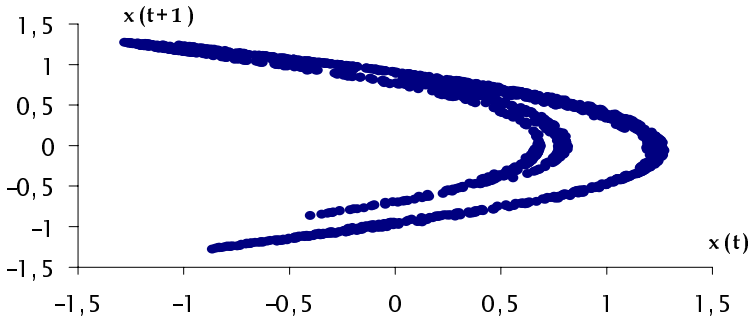


Figure 23: The predicted Henon's attractor: it was built on 1500 predicting iterations in the embedding space

Let's finally summarize the overall approach to time series processing by using only observable data. The global purpose of this approach is to identify the chaotic behavior, predict the time series at level of the individual points, and reconstruct the system dynamics. The time series of observations is represented by $\mathbf{X}(t) = (X_1(t), X_2(t),\ldots, X_n(t))$ – or shortly $\mathbf{X}(t) = X_i(t)$ – with $t = \overline{1, p}$. The time series processing approach is as follows:

1. Select any data from a single observable $X_i(t)$, $t = \overline{1, p}$.
2. Compute the embedding delay τ and take the time series by using this embedding delay.
3. Compute the minimum embedding dimension D.
4. Build the multilayer perceptron having $k \geq D-1$ input units, l hidden units, and one output unit.
5. Prepare the training data:
   $\mathbf{X}(t) = (X_i(t), X_i(t+1),\ldots, X_i(t+k-1))$
   $Y(t) = X_i(t+k)$,
   where $\mathbf{X}(t)$ is the input sequence and $Y(t)$ is desire output, for $t = \overline{1, p}$.
6. Train the neural network by using an efficient version of the backpropagation algorithm.
7. Compute the highest Lyapunov's exponent by using the neural network and identify the chaotic behavior of the nonlinear system.
8. If the n observable time series are known, where $n = \frac{D-1}{2} + 1$, the Lyapunov's spectrum can be computed.

9. Forecast the data $X_i(t)$ for the subsequent iterations.
10. Reconstruct the system dynamics.

This algorithm has a low computational complexity with respect to the approaches available in the literature and can be effectively used even with a small set of training data.

6.10. Conclusion

In this chapter the fundamental aspects of chaotic time series processing have been addressed, namely determination of the embedding parameters, the Lyapunov's spectrum, forecasting of chaotic data at the level both of the individual data points and the emergent structure. Both conventional and neural network approaches have been analyzed for chaotic signal processing. In various domains neural networks have been shown powerful tools with respect to conventional techniques. The neural approaches allow for evaluating the Lyapunov's spectrum and for reconstructing the state space accurately and efficiently only by using the observed data. Besides, the largest Lyapunov's exponent and the Lyapunov's spectrum can be computed by neural networks even on small data sets; this allows both for reducing the computationally complexity and for limit the observation time.

References

[1] P. Grassberger and I. Procaccia, Measuring the strangeness of strange attractors, Physica D 9,1983

[2] N.H. Packard, J.P. Crutchfield, J.D. Farmer and R.S. Shaw, Geometry from a Time Series, Physical Review Letters 45, 1980, pp.712-716.

[3] F. Takens, Detecting strange attractors in turbulence, Lecture Notes in Mathematics, Vol. 898, Springer-Verlag, Berlin, 1980, pp. 366-381; and in Dynamical System in Turbulence, Warlock, 1980, eds. D. Rand and L.S. Young.

[4] S. Haykin, Signal processing: Where physics and mathematics neet, IEEE Signal Processing Magazine, vol.18, pp.6-7, July 2001.

[5] S. Haykin, Adaptive filter theory, 4th Edition, Prentice-Hall, 2001.

[6] S. Haykin, Neural Networks: A comprehensive foundation, Second edition, Prentice-Hall, 1999.

[7] CybenKo G.: Approximation by Superpositions of a Sigmoidal Function, Math Control Signals Syst, 2, pp.303-314, 1989.

[8] Hertz J.A., Palmer R.G., Krogh A.S., Introduction to the theory of neural computation, Addison-Wesley, Redwood City, 1991.

[9] Hornik K., Stinchcombe M., White H., Multi-layer feedforward networks are universal approximators, Neural Networks, 2 pp.359-366, 1989.

[10] Waibel A., Consonant Recognition by Modular Construction of Large Phonetic Time-delay Neural Networks, in Touretsky D.: Advances in Neural Information Processing System, Moggzn Kaufmann, Los Altos, CA, pp. 215-223, 1989.

[11] Donoho D.L., Johnstone I.M., Kerkyacharian C., Ricard D., Wavelet shrinkage: asymptopia? Journal of the Royal Statictical Society, Series B, 57, pp.301-337, 1995.

[12] Gonzalez R., Wintz P., Digital image processing, Reading, MA: Addison-Wesley, 1987.

[13] A. Hyvarinen, E. Oja, Independent component analysis: algorithms and applications, Neural Networks, vol.13, pp.411-430, 2000.

[14] Oja E., Neural networks, principal components and subspaces, International Journal of Neural Systems, 1, pp.61-68, 1989.

[15] A.M. Albano, J. Muench, C. Schwartz, A.I. Mees and P.E. Rapp, Syngular-Value Decomposition and the Grassberger-Procaccia Algorithm, Physical Review A 38, 1988, pp. 3017-3026.

[16] M. Casdagli, S. Eubank, J.D. Farmer and J. Gibson, State space reconstruction in present of noise, Physica D 51, 1992, pp. 52-98.

[17] X. Zeng, R. Eykholt and R.A. Pielke, Estimating the Lyapunov-Exponent Spectrum from shot Time Series of Low Precision, Physical Review Letter 66, 1991, pp. 3229-3232.

[18] J. Holzfuss and G. Mayer-Kress, An approach to error estimation in the applications of dimensional algorithms, in Dimensions and Entropies in Chaotic Systems, editor G. Mayer-Kress, Springer-Verlag, New York, 1986, pp. 114-122.
[19] M.T. Rosenstein, J.J. Colins, C.J. De Luca, Reconstruction expansion as a geometry-based framework for choosing proper delay time, Physica D 73, 1994, pp. 82-98.
[20] A.M. Fraser and H.L. Swinney, Independent coordinates for strange attractor from mutual information, Physical Review A 33, 1986, pp. 1134-1140.
[21] C.E. Shannon and W. Weawer, The mathematical theory of information, University Press, Urbana Ill.
[22] H.D.I. Abarbanel, R. Brown, J. Sidorovich and L. Tsimring, The analysis of observed chaotic data in physical systems, Reviews of Modern Physics, Vol. 65, •4, 1993, pp. 1331-1392.
[23] R. Castro, T. Sauer, Correlation dimension of attractor through interspike intervals, Physical Review E 55, 1997.
[24] M.B. Kennel, R. Brown and H.D.I. Abarbanel, Determining embedding dimension for phase-space reconstruction using a geometrical construction, Physical Review A 45, 1992, pp. 3403-3411.
[25] D. Kugiumtzis, State Space Reconstruction Parameters in the Analysis of Chaotic Time Series – the Role of the Time Window Length, 1996.
[26] M. Otani and A.J. Jones, Automated embedding and the creep phenomenon in chaotic time series, 2000.
[27] A. Stefansson, N. Koncar and A.J. Jones, A note on the Gamma test, Neural Computing and Aplications 5, 1997, pp. 387-393.
[28] G. Benettin, L. Galgani, J.-M. Strelcyn, Kolmogorov entropy and numerical experiments, Physical Review A 14, 1976, pp. 2338-2345.
[29] G. Benettin, L. Galgani, A. Giorgilli, J.-M. Strelcyn, Lyapunov characteristic exponents for smooth dynamical systems and for Hamiltonian systems: A method for computing all of them. P. I: Theory. P. II: Numerical applications, Meccanica, Vol. 15, 1980, pp. 9-30.
[30] V. Golovko, Y. Savitsky, N. Maniakov and V.Rubanov, Some Aspects of Chaotic Time Series Analysis, Proceedings of the 2nd International Conference on Neural Networks and Artificial Intelligence, October 2-5, 2001, Minsk, Belarus, pp. 66-69.
[31] H. Schuster. Deterministic chaos. An introduction. Physic-Verlag, Weinhheim, 1984, p.240.