Application of Neural Network Techniques to Chaotic Signal Processing

5

Vladimir Golovko, Nikolay Manyakov and Alexander Doudkin

Laboratory of Artificial Neural Networks, Brest State Technical University,
Moskovskaja str. 267, 224017 Brest, Republic of Belarus

In this paper, we discussed the possibility of applying neural network techniques to chaotic signal processing. Among the topics under discussion were a time series analysis, the identification of chaotic behavior, forecasting and a dynamical reconstruction. The fundamental aspects of chaotic time series processing were considered, namely, the determination of embedding parameters, Lyapunov's spectrum, forecasting of chaotic data at the level of individual points as well as an emergent structure. Both conventional and neural network approaches were analyzed for chaotic signal processing. Neural networks were shown as a powerful tool in comparison with conventional techniques in various domains. The neural net approaches allowed us to evaluate Lyapunov's spectrum and to reconstruct state space accurately and efficiently by using some observed data. In addition, it was found that Lyapunov's highest exponent and Lyapunov's spectrum could be computed by means of neural networks even on small data sets; these results allowed us to reduce a computation complexity and to limit observation time.

Keywords: Dynamical system, chaos theory, embedding parameters, attractor, Lyapunov exponents, neural networks.

1. BASIC DEFINITIONS

A dynamical system is regarded by us as a system of any nature (physical, chemical, biological etc.), which can be described in some mathematical form: by the differential or difference equations. Changes in the state of a dynamical system are called the evolutions of a dynamical system and the equations describing these changes are called evolution equations. Variables, which present in evolution equations and which describe state changes of a dynamical system, are called phase coordinates. The set of phase coordinates, or in other words the space in which a dynamical system exists, is called phase space.

A system evolution greatly depends on the presence or absence of the internal energy loss. For example, if we deal with a dynamical system of mechanical nature, the internal energy loss can arise because of friction. The internal energy loss means the decrease of the value of the object (the measure in phase space) under the influence of the system evolution throughout time. The dynamical systems, which have a continuous loss of internal energy, are called dissipative, the systems, in which the internal energy remains unchanged, are called conservative or Hamilton's systems.

The subject of the study in a chaos theory is called chaotic systems. The system is called chaotic if it has a long-term aperiodic behavior and a sensitive dependence on the initial conditions. Long-term aperiodic behavior is regarded as the absence of periodic and quasi-periodic orbits, and of the orbit, which is attracted to a stable point. Sensitive dependence on the initial conditions means that two nearby points of the phase space have trajectories, which disperse throughout time [6]. So a little inaccuracy will give us a substantially different behavior.

Lyapunov's exponent is used as a measure of exponential divergence of the nearest trajectories [7,8]. When a dynamical system has positive Lyapunov's highest exponent, we can conclude that the system is chaotic. If we consider the divergence of trajectories in every direction of phase space basis, we can speak about several Lyapunov exponents, which compose Lyapunov's spectrum.

Any dissipative dynamical system has an attractor, i.e., a compact subset of phase space, which asymptotically attracts all the nearest trajectories [8]. In the case of a chaotic system, we have a complex geometrical object with fractal dimension as an attractor. Such types of attractors are called strange attractors.

The basic problem in the analysis of a chaotic time series consists in finding such parameters as embedding dimension and time delay.

2. BASIC EXAMPLES OF CHAOTIC SYSTEMS

In this paper, we will examine all proposed algorithms in the following dynamical system [2], which under suitable conditions has chaotic behavior.

We take Henon's system as a dynamical system, which is described by the following difference equations:

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

where $\alpha = 1.4$ and $\beta = 0.3$ for chaotic behavior. An attractor of this system and its *X*-series are presented in Figs. 1 and 2.

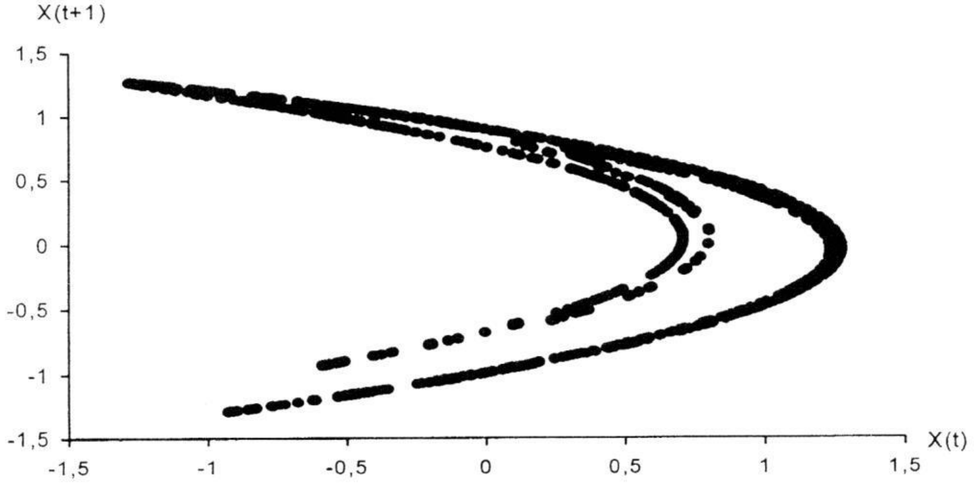

Fig. 1. Henon's original attractor.

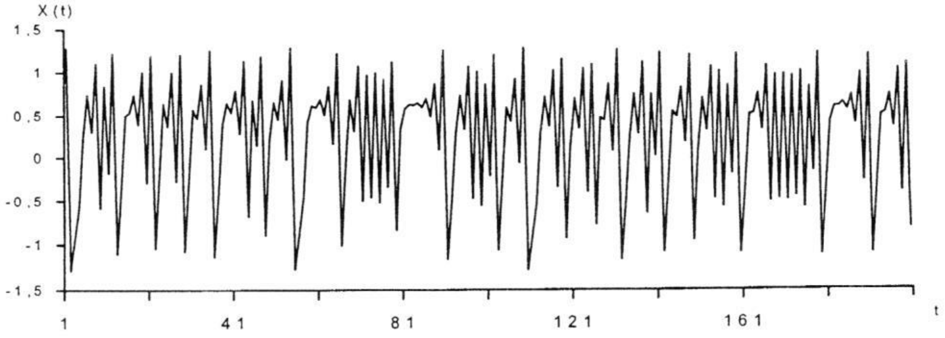

Fig. 2. Henon's *X*-series (first 200 elements).

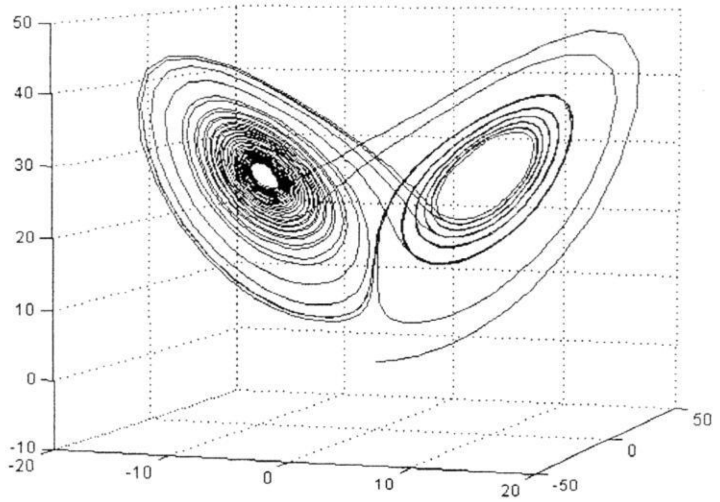

Fig. 3. Lorenz's original attractor.

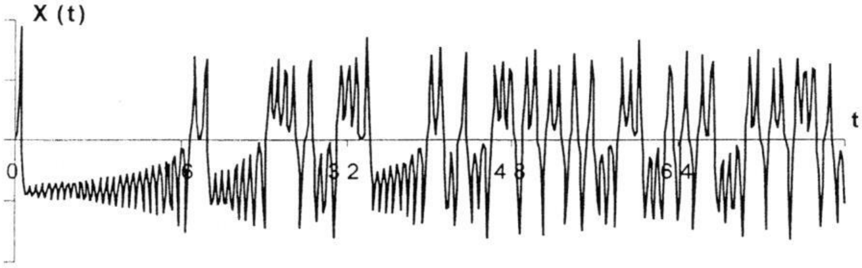

Fig. 4. Lorenz's *X*-series.

We consider Lorenz's and Roessler's systems as chaotic dynamical systems, which are described by the system of differential equations.

Lorenz's system is described by the following system:

$$\begin{cases} \dfrac{dx}{dt} = \sigma(y - x) \\ \dfrac{dy}{dt} = rx - y - xz \\ \dfrac{dz}{dt} = yz - bz \end{cases}$$

which becomes chaotic at $\sigma = 10$, $b = 8/3$ and $r = 28$. An attractor of this system and its *X*-series are presented in Figs. 3 and 4.

Roessler's dynamical system is the following system:

$$\begin{cases} \dfrac{dx}{dt} = -y - z \\ \dfrac{dy}{dt} = x + ay \\ \dfrac{dz}{dt} = b + (x - r)z \end{cases}$$

which becomes chaotic at $a = b = 0.2$, $r = 5.7$.

3. TIME SERIES ANALYSIS

Let us consider a certain dynamical system. We can observe only one phase variable of this system. Let us illustrate the changes of this phase variable in the form of the quantized time series:

$$x(1),\ x(2),\ldots,\ x(N)\,,$$

where N is the number of measures.

The basic idea of the dimension estimation of the initial chaotic system attractor from this one-dimensional time series is based on the reconstruction of the initial attractor in pseudo-phase space.

Pseudo-space reconstruction is a mapping, where every point $x(t)$ of the time series is associated with point $(x(t), x(t+\tau),\ldots, x(t+(D-1)\tau)) \in R^D$, where t is quantized time ($t = \overline{1,\ N - D \cdot \tau}$), τ is time delay and D is the embedding dimension.

Takens [8] shows that by using only a one-dimensional observation, we can construct in pseudo-phase space an attractor $(x(t), x(t+\tau),\ldots, x(t+(D-1)\tau))$, which preserves the basic topological and dynamical properties of the attractor of the initial dynamical system. For this purpose, we must take an embedding dimension as $D = 2[d] + 1$, where $[d]$ denotes the integer part of the attractor fractal dimension d.

So we can conclude that the analysis of the chaotic process consists in finding a dynamical system embedding parameter, such as a time delay [6,13-15] and an embedding dimension [9-12,16-18], for good pseudo-phase reconstructions. Those parameters are also necessary for the best prediction of a chaotic process, as well as for the choice of the best network architecture (it will be shown in the following sections).

4. ESTIMATION OF TIME DELAY

For a time delay τ estimation, we can use the following approaches [6,13-15]:

- The method of autocorrelation function;
- The method of mutual information;
- The average displacement method.

The first two of these methods use the fact that for good phase reconstruction, we must take the most possibly independent pseudo-phase coordinate. We must also keep in mind that the time delay must be of the lowest value. In this way, we receive more points which construct an attractor in pseudo-space.

4.1 The Method of Autocorrelation Function

Let us consider autocorrelation functions

$$R(\tau) = \frac{1}{N \cdot (N-\tau)} \cdot \sum_{t=1}^{N-\tau} \dot{x}(t) \cdot \dot{x}(t+\tau)\,,$$

where $\dot{x}(t) = x(t) - E\{x\}$ is a centered version of a time series.

This function gives us a measure of correlation between time series, which differ in the time τ. This refers to the amount of knowledge on the values of this series that we can receive from the current values throughout the time τ. However, we must note that this knowledge is only linear.

In this case, we can take an optimal time delay as the first zero value (or the first minimum) of the autocorrelation function. This method is very simple and does not require much calculation. As we search for a noncorrelation, we cannot be sure that we will receive independent coordinates. So we do not estimate an optimal time delay τ.

4.2 The Method of Mutual Information

Let us divide a value area into L intervals, where $L = [\log_2 N] + 1$. Let us denote as A_i the event where the time series value is located in the i-th interval, and as B_j where the event in which subsequent observation through time τ falls in the j-th interval. After that we can calculate the mutual information function using the formula

$$I(\tau) = -\sum_{i=1}^{L} \sum_{j=1}^{L} P(A_i \cdot B_j) \cdot \log_2 \frac{P(A_i \cdot B_j)}{P(A_i) \cdot P(B_j)}\,,$$

where $P(.)$denotes an event rate.

On one hand, the mutual information function is a more exact measure of independence, and, on the other hand, it is more complicated than the autocorrelation function. In this method we take the first minimum of the mutual information function $I(\tau)$ as the optimal time delay τ.

To make calculation easier, we can take a modified version of this formula:

$$I_1(\tau) = -\sum_{i=1}^{L}\sum_{j=1}^{L}[P(A_i \cdot B_j) - P(A_i)\cdot P(B_j)].$$

This function has the "same shape" as $I(\tau)$. It means that these functions have the same intervals of growth and falling, and the same bending points. In practice, when we use formula $I_1(\tau)$ instead of $I(\tau)$, we shorten the time of calculation by 10-15%.

As we see from Fig. 5 we receive the first minimum for the mutual information function for Lorenz's X-series in $\tau = 16$ (0.16), and the first nearest to zero value of autocorrelation function $\tau = 47$ (0.47).

4.3 The Average Displacement Method

This method was first proposed by Rosenstein [13] in 1994. It is based on the estimation of the average expansion of the reconstructed attractor from the identity line of the reconstructed phase space. The function of the average expansion is defined as

$$S_D(\tau) = \frac{1}{N - D\cdot\tau}\cdot\sum_{t=1}^{N-D\cdot\tau}\sqrt{\sum_{k=1}^{D}[x(t+k\cdot\tau) - x(t)]^2},$$

where D is the embedding dimension concerned in this step.

It is evident that for $\tau = 0$ the reconstructed trajectory is a subset of the identity line of the reconstructed phase space. By increasing time delay τ, the reconstructed trajectory moved away from the identity line and

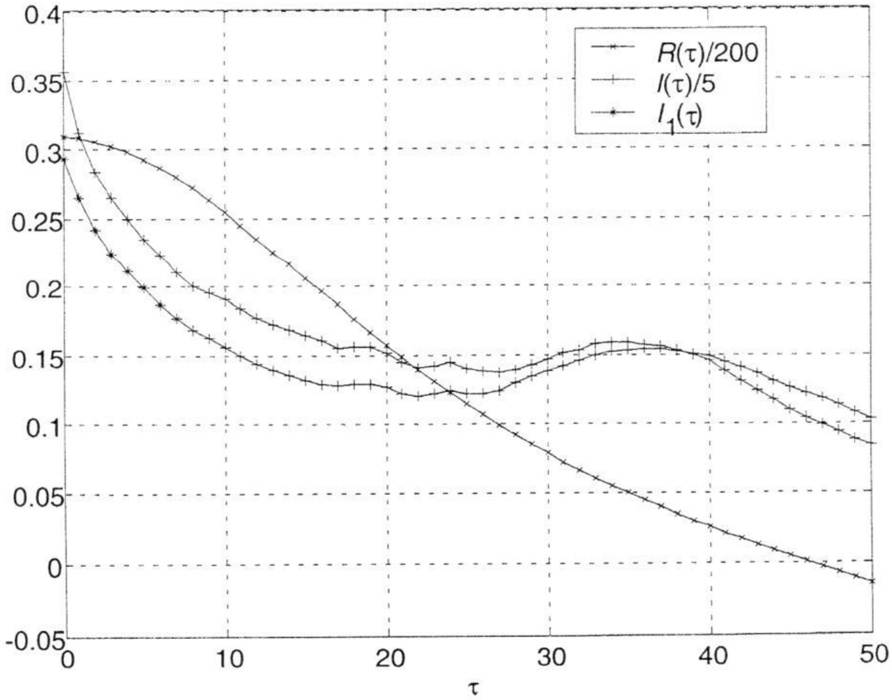

Fig. 5. Graphs of autocorrelation function $R(\tau)$, mutual information $I(\tau)$and modified mutual information $I_1(\tau)$ of Lorenz's X-series with quantized step 0.01.

functions $S_D(\tau)$go to satiation. The τ value is considered as an optimal time delay when the satiation begins. For the highest value of the embedding dimension D, the function $S_D(\tau)$goes to satiation with a low value of τ. So we can see that this method is effective in the cases when the embedding dimension D is *apriori* known.

5. ESTIMATION OF EMBEDDING DIMENSION

For the estimation of the embedding dimension of the reconstructed phase space, we can use one of the following methods [9-12,16-18]:

- Correlation dimension method;
- False nearest-neighbors method;
- Gamma test.

5.1 Correlation Dimension Method

Let us consider the conception of fractal dimension, and some bounded set F in the space R^n. For every $\varepsilon > 0$ we consider a finite covering of this set by an n-dimensional cube with edges equal to ε. If we define the number of such coverings as $N(\varepsilon)$, we have the dependence in the following form

$$N(\varepsilon) \cong \varepsilon^{-D_0} .$$

From this equality, we can receive a formula of calculation of the set capacity D_0:

$$D_0 = -\lim_{\varepsilon\to 0} \log_\varepsilon N(\varepsilon) .$$

We can also define some other dimensions, such as Hausdorff dimension d_H, the information dimension d_I, the correlation dimension D_2 and others. However, the value of these dimensions do not differ in the problems of chaos theory and nonlinear dynamic. So we will simply speak about fractal dimension and denote it as d.

To estimate the fractal dimension, we can use Grassberger-Procaccia's algorithm, which calculates the correlation dimension D_2 of the set F by the extracts $\{\overline{x_i}\}_{i=\overline{1,M}}$ from this set.

Let us cover the set F by the cells of the same size ε. Let us denote as p_i the probability of the event which arbitrary point x_j of the set F is located in the i-th cell. So the correlation dimension can be calculated as

$$D_2 = \lim_{\varepsilon\to 0} \frac{\ln C(\varepsilon)}{\ln \varepsilon} ,$$

where the correlation integral $C(\varepsilon)$ is estimated according to

$$C(\varepsilon) = \lim_{M\to\infty} \frac{1}{M(M-1)} \sum_{i=1}^{M}\sum_{j=1}^{M} \theta\left(\varepsilon - \left\|\overline{x_i} - \overline{x_j}\right\|\right),$$

where $\theta(\cdot)$ is the Heaviside function,

$$\theta(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} .$$

To estimate the correlation dimension according to Grassberger-Procaccia's algorithm, we must plot the diagram of $\ln(C(\varepsilon))$versus $\ln \varepsilon$. The slope d of the linear part of the received curve estimates the correlation dimension D_2.

When we must calculate the correlation dimension of the attractor in an m-dimensional reconstructed phase space from the single one-dimensional time series, we follow the following method. After the estimation of the time delay τ, we make the reconstruction of the phase space of the dimension m. So we receive the set of points

$$\overline{x_i} = (\, x(i),\, x(i+\tau), \ldots,\, x(\, i + (m-1)\tau\,)\,) \in R^m ,$$

where $t = \overline{1,\ N - m \cdot \tau}$. For this extract we calculate $D_2(m)$ according to the above algorithm. After the estimation of the dimension for $m = 1, 2, 3\ldots$ we can see that the value of $D_2(m)$ becomes stable with the increase in the embedding dimension m. This stable value is an estimation of the correlation dimension of the attractor in the reconstructed phase space. So using Takens theorem, we can calculate the embedding dimension by the formula $D = 2[D_2] + 1$. If we do not have a satiation of the dimension estimation $D_2(m)$ with the increase in m, we can conclude that there is some noise.

5.2 False Nearest-neighbors Method

From Takens' embedding theorem we know that for appropriate τ and D, we have a topologically equivalent (homeomorphic) initial and reconstructed attractors. As we do not have intersections of the trajectories in the initial attractor, we can conclude that the reconstructed attractor also does not have self-intersections. The self-intersection of the reconstruction attractor indicates that we do not have an appropriate embedding dimension yet, and we must increase the dimension. The condition of non-self-intersection is the case, when all neighbor points in R^m should be also the neighbors in R^{m+1}. The false nearest-neighbors method allows us to estimate the dimension D, which has a small percentage of false nearest-neighbors (such a pair of points, which are the nearest in the reconstructed phase space of dimensional D, and are not the same in dimension $D + 1$). The dimension D received estimates an appropriate embedding dimension, where we have a reconstructed attractor without self-intersections.

The algorithm of the false nearest-neighbor method consists of the following steps:

1. Let $m = 1$.
2. For every point $\overline{x_i}$ in an m-dimensional reconstructed phase space, we look for the nearest neighbor $\overline{x_j}$.
3. We calculate the distance $\left\|\overline{x_i} - \overline{x_j}\right\|_m$ using Euclid's metric.
4. We calculate the distance $\left\|\overline{x_i} - \overline{x_j}\right\|_{m+1}$ between the images of this point in the $(m + 1)$—dimensional reconstructed phase space and estimate the ratio

$$R_i = \sqrt{\frac{\left\|\overline{x_i} - \overline{x_j}\right\|_{m+1} - \left\|\overline{x_i} - \overline{x_j}\right\|_m}{\left\|\overline{x_i} - \overline{x_j}\right\|_m}} = \frac{\left|x(i + m \cdot \tau) - x(j + m \cdot \tau)\right|}{\left\|\overline{x_i} - \overline{x_j}\right\|_m}.$$

5. In the case, when $R_i > R_t$, where R_t is an appropriate threshold, we denote $\overline{x_j}$ as a false nearest-neighbor regarding point $\overline{x_i}$.
6. After the calculation of the number P of such false nearest-neighbors for every point $\overline{x_i}$ of the reconstructed phase space, we estimate P/N.
7. Repeat the algorithm beginning with step 2.
8. If the ratio P/N is sufficiently small, we stop the algorithm.

The recommended value of R_t is equal to 2.

The graphs for the estimation of embedding dimension for Henon's and Lorenz's series, respectively, are presented in Figs. 6 and 7. From these figures, we can conclude that the appropriate embedding dimension for Henon's data is equal to 3, and for Lorenz's data, it is equal to 5.

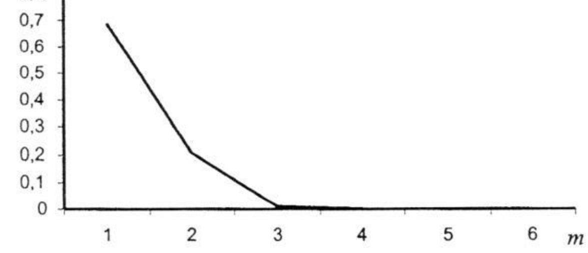

Fig. 6. The estimation of the embedding dimension from Henon's *X*-series.

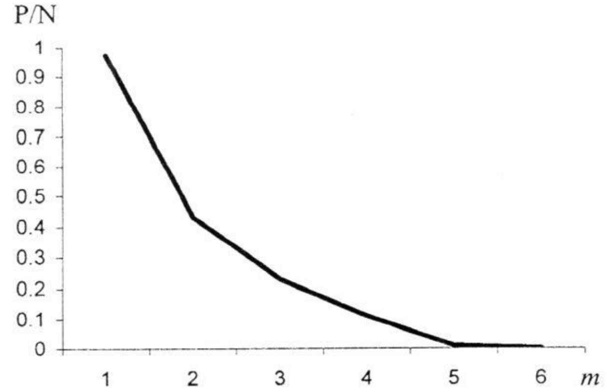

Fig. 7. The estimation of the embedding dimension from Lorenz's *X*-series

5.3 Gamma Test

Let us have a continuous mapping f: $R^m \to R$. Suppose we have y, which is described by the formula

$$y = f(x_1, x_2, \ldots, x_m) + r,$$

where r is a random component (it may arise because of some noise). The main purpose of the gamma test is an estimation of the r variance, i.e., the estimation of a low bound of a square error of the calculation of variable y. Let us have the set of m-dimensional vectors $X = \left\{\overline{x_i} \in R^m, i = \overline{1,M}\right\}$. We associate every vector with value y_i described by the above formula. Let us denote as $\overline{x_{i_p}}$ the p-th nearest neighbor of $\overline{x_i}$ in the given set X, and as y_{i_p} the value that corresponds to $\overline{x_{i_p}}$. Let us consider the following functions:

$$\Delta(p) = \frac{1}{p}\sum_{h=1}^{p}\frac{1}{M}\sum_{i=1}^{M}\left\|\overline{x_i} - \overline{x_{i_p}}\right\|_m^2$$

and

$$\Gamma(p) = \frac{1}{p}\sum_{h=1}^{p}\frac{1}{2M}\sum_{i=1}^{M}\left(y_i - y_{i_p}\right)^2.$$

For every $p = \overline{1, p_{\max}}$ (the value of $p_{\max}$ is usually taken from the interval from 20 to 50) we calculate the pairs $(\Delta(p), \Gamma(p))$, for which we can recognize the linear dependence in the form of a regression line $\Gamma = A \cdot \Delta + \overline{\Gamma}$. The coefficients A and $\overline{\Gamma}$ are estimated according to the least-squares method. The coefficient A received characterizes the complexity of the surface $y = f(x_1, x_2, \ldots, x_m) + r$ and the value of $\overline{\Gamma}$ is a desired value of the gamma test.

Let us consider the gamma-test implementation [18] on the estimation of an embedding dimension. Let us know the time changes only of one phase variable. After the quantization we have a time series $x(t)$. And after the phase-space reconstruction from a single one-dimensional time series $x(t)$, we receive the set of m-dimensional vectors

$$\overline{x}(t) = (x(t), x(t+\tau), \ldots, x(t + (m-1)\tau)),$$

where τ is defined as an earlier estimated time delay.

Let us define mapping f from the set of vectors $\overline{x}(t)$ to space R as

$$f\left(\overline{x}(t)\right) = x(t + m \cdot \tau).$$

We apply the gamma test to this mapping and estimate $\overline{\Gamma}$. It is evident that for any dimension m we have different values of $\overline{\Gamma}$. So we can speak about the dependence of $\overline{\Gamma}$ from m, i.e., from the function $\overline{\Gamma} = \overline{\Gamma}(m)$. The experiments show that this function with m increasing usually decreases and then increases. The value of m, in which $\overline{\Gamma}(m)$ reaches its minimal value, is recommended to be taken as an embedding dimension, i.e.,

$$D = \arg\min_{m}\left\{\overline{\Gamma}(m)\right\}.$$

There are situations when $\overline{\Gamma}(m)$ increases or decreases monotonically. The cause of such behavior may be due to a bad choice of the time delay or due to the high level of noise in the input data.

6. PREDICTION AND RECONSTRUCTION OF THE CHAOTIC PROCESS

The aim of time series forecasting can be described in the following way [1-3]: for the initial sequence $x(1)$, $x(2),\ldots, x(N)$ we must find its continuation $x(N + 1)$, $x(N + 2),\ldots$ For this purpose we use a nonlinear model $x(t + k \cdot \tau) = F(x(t), x(t + \tau),\ldots, x(t + (k - 1)\tau))$, where F is the nonlinear function constructed with the use of a neural network, k is the size of slide windows. A multilayer feedforward neural network, which is well known for its approximation properties is taken as a neural network.

As mentioned above, we must take an appropriate time delay τ and an embedding dimension D for the reconstruction of the system's attractor. In this case, we reconstruct an attractor in phase space R^D and the orbits do not have self-intersections in it. The reconstructed attractor also preserves the property, so that we can unambiguously calculate the latter for any $(D - 1)$ phase coordinate. Taking this into account we can conclude, that we must construct a neural network, which has $k > D - 1$. In this case, we received training samples in the following form

$$Y = x(i + k \cdot \tau),$$
$$X = (x(i), x(i + \tau),\ldots, x(i + (k - 1) \cdot \tau)),$$

where $i = \overline{1, N - k \cdot \tau}$, Y is an output value of a neural network, and X is an input value for the corresponding time moment.

After training this neural network may forecast a chaotic signal. So we must construct neural networks with $k \geq D - 1 = 5 - 1 = 4$ neurons in an input layer and $\tau = 0.16$ for Lorenz's X-series, and with $k \geq D - 1 = 3 - 1 = 2$ and $\tau = 1$ for Henon's X-series. These embedding parameters were calculated using the above mentioned technique.

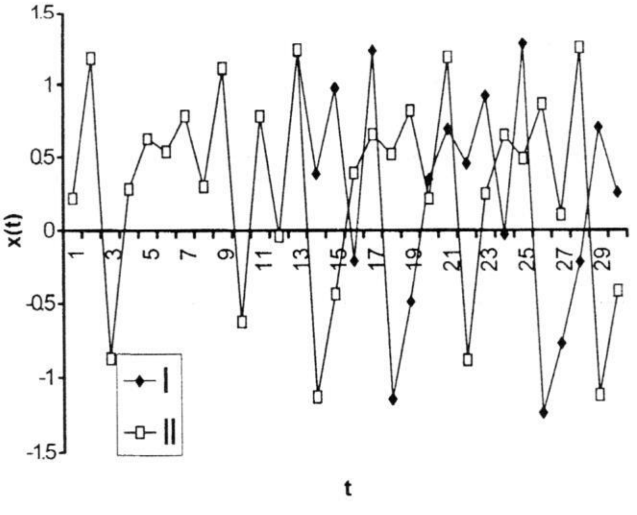

Fig. 8. Henon's process. The prediction results for 30 forecasting iterations: I – the prediction, II – the original time series.

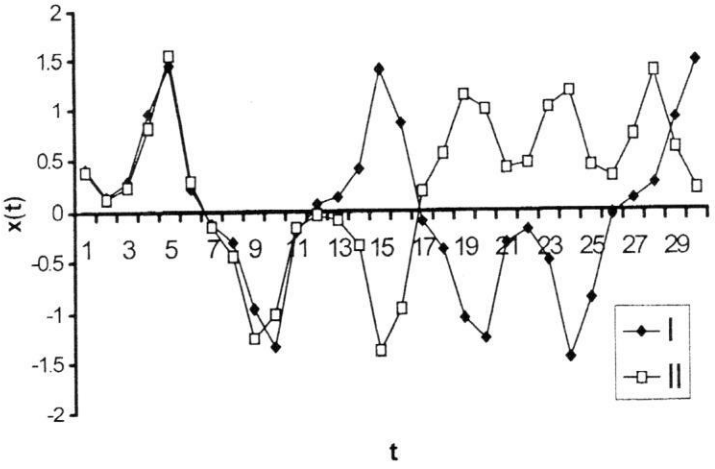

Fig. 9. Lorenz's process. The prediction results for 30 forecasting iterations: I – the prediction, II – the original time series.

In our experiments we take a multilayer perceptron with 7 units in an input layer, 5 sigmoid units in a hidden layer and one linear unit in an output layer. We take 930 training samples for Henon's *X*-series and 800 samples for Lorenz's *X*-series.

Figures 8 and 9 present the results of Henon's and Lorenz's series prediction in 30 steps. As we can see, the forecasting accuracy decreases when time increases. That is because of the chaotic nature of the initial series. In a general case, the prediction horizon *T* for a chaotic time series can be estimated by

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

where $K = \sum_i \lambda_i$ is Kolmogorov's entropy, $\lambda_i > 0$ is positive Lyapunov exponents of a chaotic system, d_0 is the initial prediction error.

On the other hand the network, constructed and trained as it was described above, can preserve the system's dynamic. For example, in Fig. 10, we can see a predicted attractor of Henon's *X*-series in a reconstructed phase space.

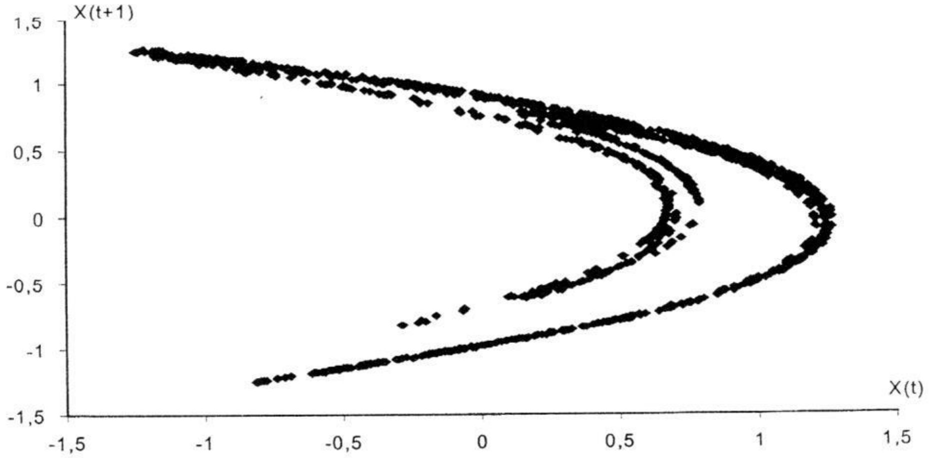

Fig. 10. Henon's predicted attractor: it was built on 1500 forecasting iterations in the reconstructed phase space.

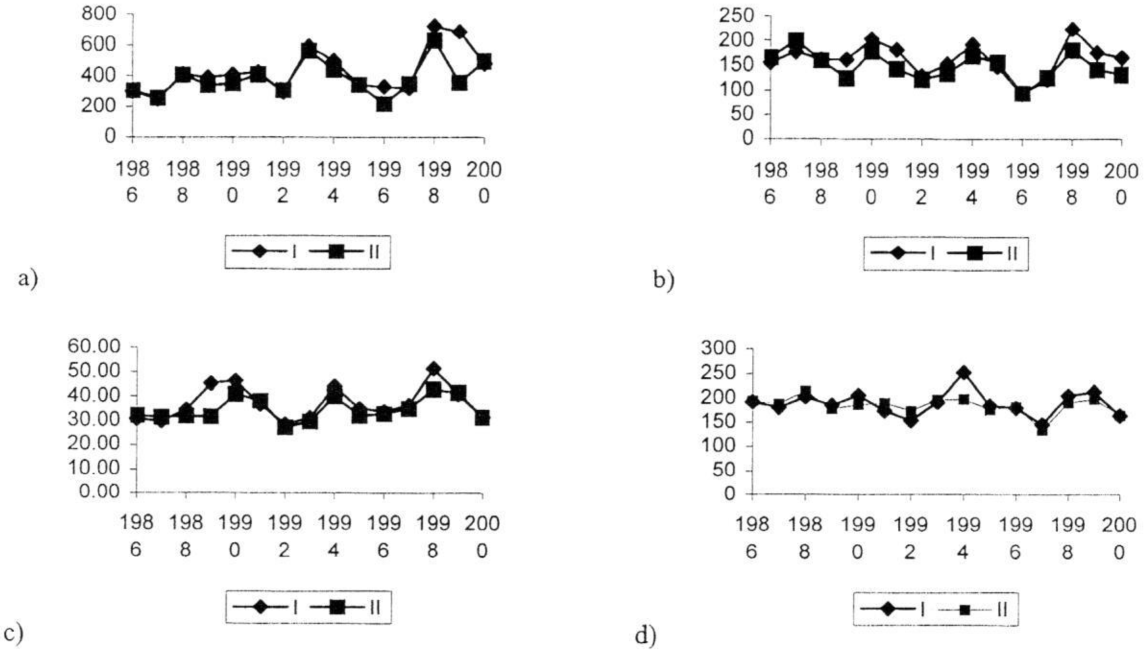

Fig. 11. The results of the numerical experiment on forecasting annual river flow with one year forecast: a – the river Pripiat', b – the river Dnepr, c – the river Berezina, d – the river Neman: I – the original time series, II – the prediction.

7. FORECASTING OF REAL NATURE DATA

Let us examine the problem of prediction of a long-term oscillation of annual river flow. The problem of its analysis and long-term forecasting is one of the actual and complex problems in hydrology. The importance of a long-term prediction is determined by the fact that if we reach the defensible level of 70% we attain a new level of water sources management. The results of numerous investigations show that the long-term oscillation of annual river flow is subject to notable changes, which are caused by natural-climatic and anthropogenic influences.

For hydrologic characteristic forecasting (in particular the prediction of annual river flow in advance), the methods using simple and complex Markov's chains and a dynamic-statistical method of Alehin are widely used at present. The results obtained by applying these methods only seemed to be good. When we used Markov's chains, we received random points, which were distributed along a horizontal trend (because we only consider a stationary series) with variance, asymmetry and autocorrelation as in the initial series. So the closeness of the initial and received points was random. When we used Alehin's methods, we used a linear prediction model. So we received a forecast, which is approached on a horizontal trend. A large quantity of points, which get into a small forecasting error, depends on a small amplitude of oscillation related to an average value. The use of this method for forecasting of a time series with a large amplitude of oscillation is not possible. The method based on a neural network technique allows us to overcome all these problems.

For the investigation of using hydrological characteristics forecasting by a neural network technique we used annual river flow with the longest observational series (the rivers Pripiat' (Mozyr'), Neman (Grodno), Berezina (Bobrujsk), amd Dniepr (Rechitsa)). These data were given by the Polessie Problem Department of the National Academy of Sciences of Belarus. The numerical experiment was carried out within the period of 15 years from 1986 to 2000 with an advanced time forecast of 1, 2, 3, 4 and 5 years. We took the measurements of the annual river flow up to 1986 as a training set. The network architecture was taken according to the proposed method above. For every time series we calculated an embedding dimension and a time delay. We received the next embedding parameters: for the river Pripiat' – time delay $\tau = 2$ and embedding dimension $D = 5$, for the river Neman — $\tau = 4$ and $D = 3$, for the river Berezina — $\tau = 3$ and $D = 4$, for the river Dnepr — $\tau = 3$ and $D = 4$.

Table 1. The results of the numerical experiment on forecasting of annual river flow with different advance time.

Interval of forecasting error, %	Forecasting error									
	1 year		2 year		3 year		4 year		5 year	
	Number of years	%	Number of years	%	Number of years	%	Number of years	%	Number of years	%
The river Pripiat' (Mozyr')										
0-5	8	53.3	12	41.4	15	35.7	15	27.8	16	24.6
0-10	9	60.0	15	51.7	18	42.9	19	35.2	21	32.3
0-15	13	86.7	19	65.5	24	57.1	26	48.1	29	44.6
0-20	13	86.7	22	75.9	28	66.7	30	55.6	34	52.3
Total	15		29		42		54		65	
The river Dniepr (Rechitsa)										
0-5	4	26.7	6	20.7	7	16.7	9	16.7	12	18.5
0-10	6	40.0	11	37.9	13	31.0	16	29.6	22	33.8
0-15	10	66.7	15	51.7	19	45.2	23	42.6	29	44.6
0-20	12	80.0	17	58.6	23	54.8	28	51.9	35	53.8
Total	15				42		54		65	
The river Berezina (Bobrujsk)										
0-5	8	53.3	12	41.4	15	35.7	20	37.0	21	32.3
0-10	12	80.0	16	55.2	20	47.6	25	46.3	27	41.5
0-15	13	86.7	20	69.0	26	61.9	33	61.1	36	55.4
0-20	14	93.3	22	75.9	30	71.4	38	70.4	43	66.2
Total	15		29		42		54		65	
The river Neman (Grodno)										
0-5	7	46.7	12	41.4	14	33.3	18	33.3	20	30.8
0-10	13	86.7	20	69.0	27	64.3	32	59.3	36	55.4
0-15	14	93.7	22	75.9	30	71.4	37	68.5	41	63.1
0-20	14	93.7	24	82.8	34	81.0	41	75.9	46	70.8
Total	15		29		42		54		65	
Altogether										
0-5	27	45.0	42	36.2	51	30.4	62	28.7	69	26.5
0-10	40	66.7	62	53.4	78	46.4	92	42.6	106	40.8
0-15	50	83.3	76	65.5	99	58.9	119	55.1	135	51.9
0-20	53	88.3	85	73.3	115	68.5	137	63.4	158	60.8
Total	60		116		168		216		260	

The forecasting results with one year advance time for different Belarusian rivers are presented in Fig. 11. We can see that the results of this prediction are acceptable. We can see that 45% of all forecasting points lies in the range ±5% (Table 1), and 88.3% lies in the range of ±20%. With the increase in advance time, accuracy decreased.

Table 2 shows the comparative characteristics of different forecasting methods for the prediction of the oscillation of annual Belarusian river flow.

8. LYAPUNOV'S EXPONENTS

When we consider dynamical systems, Lyapunov's exponents play an important role. They give a quantitative measure of a system's chaotic character. Any system, described by an n differential or a difference equation, has n exponents λ_i ($i = 1, 2, \ldots, n$), which are called Lyapunov's spectrum. For its estimation, we need

Table 2. The comparative results of the numerical experiment on forecasting of annual river flow with different prediction methods.

Interval of forecasting error, %	Forecasting error									
	1 year				1 year				1 year	
	Number of years		Number of years		Number of years		Number of years		Number of years	
Markov's simple chain										
0-5	0	0.0	9	7.8	15	8.9	20	9.3	23	8.8
0-10	3	5.0	21	18.1	33	19.6	44	20.4	51	19.6
0-15	11	18.3	36	31.0	53	31.5	69	31.9	80	30.8
0-20	19	31.7	47	40.5	69	41.1	90	41.7	104	40.0
Total	60		116		168		216		260	
Markov's complex chain										
0-5	5	8.3	11	9.5	14	8.3	20	9.3	26	10.0
0-10	10	16.7	23	19.8	30	17.9	37	17.1	44	16.9
0-15	14	23.3	33	28.4	46	27.4	62	28.7	75	28.8
0-20	19	31.7	43	37.1	58	34.5	79	36.6	97	37.3
Total	60		116		168		216		260	
A dynamic-statistical method of Alehin										
0-5	9	15.0	16	13.8	23	13.7	30	13.9	34	13.1
0-10	19	31.7	36	31.0	54	32.1	70	32.4	83	31.9
0-15	32	53.3	59	50.9	85	50.6	106	49.1	125	48.1
0-20	38	63.3	68	58.6	99	58.9	128	59.3	153	58.8
Total	60		116		168		216		260	
A neural network										
0-5	27	45.0	42	36.2	51	30.4	62	28.7	69	26.5
0-10	40	66.7	62	53.4	78	46.4	92	42.6	106	40.8
0-15	50	83.3	76	65.5	99	58.9	119	55.1	135	51.9
0-20	53	88.3	85	73.3	115	68.5	137	63.4	158	60.8
Total	60		116		168		216		260	

to linearize a corresponding normal system, which assigns the dynamic of the system, i.e., we must investigate the equations in its first approximation. The matrix of the received system will have an n eigenvalue. The natural logarithms of its absolute values will characterize Lyapunov's exponents. Lyapunov's spectrum [4,5,16] describes the changes of an n-dimensional volume in tangent space to the dynamical system phase space. So it characterizes a dissipation (contraction of volume), as well as sensitivity to the initial conditions (a divergence in the directions of some eigenvectors of the matrix of the first approximation). Lyapunov's highest exponent [19-24] describes the average speed of an exponential divergence of the trajectories, which are nearby in the initial moment of time. For a chaotic behavior we need Lyapunov's highest positive exponent.

If we take a sphere of a small radius in an n-dimensional phase space as an initial condition, it will be transferred into an n-dimensional ellipsoid throughout time. Lyapunov's spectrum characterizes the changes of semi-axis lengths of this ellipsoid (see Fig. 12).

Lyapunov's exponents are defined in accordance with the formula

$$\lambda_i = \lim_{t\to\infty}\left(\frac{1}{t}\cdot\ln\frac{l_i(t)}{l_i(0)}\right),$$

where $l_i(0)$ and $l_i(t)$ correspond to the length of the i-th ellipsoid semi-axis at the initial moment of time and at a time moment t, respectively.

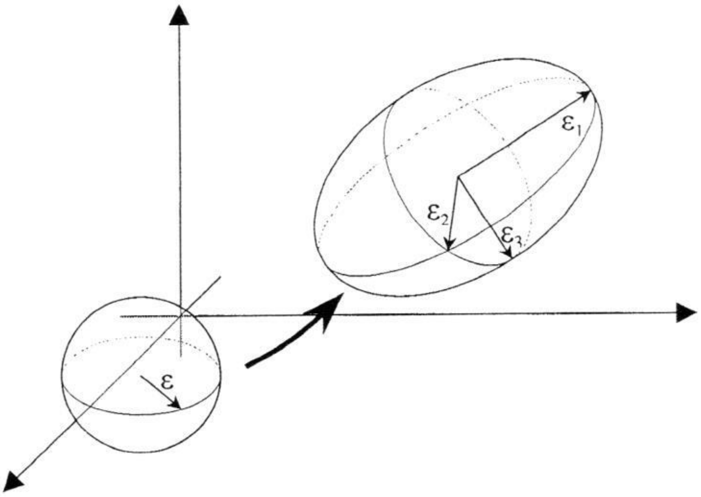

Fig. 12. Lyapunov's spectrum definition (3-dimensional case).

The estimation of Lyapunov's highest exponent is carried out in accordance with Benettin's algorithm [19] when we know the system of equations, which describes the dynamics of systems:

We take an arbitrary point in a phase space and iterate it until it gets onto the system attractor. Marking it as x_0, we take any nearest point of the attractor and mark it as y_0. So we receive a vector $\vec{w}(0) = y_0 - x_0$ with length $d(0) = \left|\vec{w}(0)\right|$. According to the system dynamic we calculate the images of these points in time t. So we receive points $\overline{x_1}$ and $\overline{y_1}$, respectively, which are situated at a distance of $d(1) = \left|\overline{w}(1)\right| = \left|\overline{y_1} - \overline{x_1}\right|$. Remarking $\overline{x_1}$ as x_1, we take y_1 as the point found in the distance $d(0)$ from x_1 in the direction of vector $\vec{w}(1)$ (i.e., $y_1 = x_1 + \frac{\overline{y_1} - x_1}{d(1)} \cdot d(0)$) we again iterate it (onto the time t) and calculate $d(2)$. By repeating this procedure, we receive the set of numbers $d(k)$, $k = \overline{1, N}$. Then Lyapunov's highest exponent can be estimated as an arithmetical average of the logarithm of ratio $d(k)$ to the initial length $d(0)$, i.e., by the formula $\lambda = \frac{1}{n \cdot t} \cdot \sum_{k=1}^{N} \ln\left(\frac{d(k)}{d(0)}\right)$. For a more accurate estimate, we must take N as a sufficiently great number, and that takes a large computational burden.

For a more rapid calculation of Lyapunov's highest exponent we can calculate logarithm changes of distance d between two nearest points x_0 and $y_0 = x_0 + d$ throughout the time. We examine only such d, which satisfy to the $d < 1$. After the construction of the regression line we determine its slope. That slope is a rough estimate of Lyapunov's highest exponent. Using this method we calculate Lyapunov's highest exponents for Henon's (0.418) and Lorence's (0.906) systems.

When we must calculate numerically Lyapunov's highest exponent from a time series of only one measurement, we do it in the following way [20]: For an initial time series we produce an embedding into the reconstructed phase space of the dimension D with the time delay τ in accordance with Takens' theorem. So we receive the orbit $[x(t), x(t + \tau), \ldots, x(t + (D - 1)\tau)]$ without self-intersections in the space R^D. Taking an arbitrary point in the reconstruct attractor, we search for the nearest point in accordance with Euclid's metrics. To have an accurate estimation of the exponent such points should be at a very close distance, not more than 10^{-8}. To find such points, we need a time series with a large number of counts, which greatly influences the checking time. After the choice is made, we examine the changes of the distance between its images throughout time. Next, using the above procedure, we calculate Lyapunov's highest exponent. To overcome this time-dependent computational problem, we will present an algorithm using a neural network.

Another big problem in nonlinear dynamics is the calculation of Lyapunov's spectrum. Its numerical calculation is based in general on Benettin's algorithm [19]. Let us have the system dynamics living in an n-dimensional phase space of some normal differential or difference equations. If we estimate Lyapunov's spectrum using the above definitions, namely, calculating the length's changes of the nearest points in the direction of tangent vectors, we do not receive an acceptable result because of the instability of these directions. This occurs due to an exponentional decrease in the angles between the tangent vectors in the process of the system dynamics. All tangent vectors will come close, throughout time, to the vector along which the changes of the highest exponent take place (namely, the above method in which the highest exponent's calculation is based on). To overcome this fact, it was proposed to orthogonalize vectors in every step with the use of Gram-Schmidt's methods.

Let us choose in the n-dimensional phase space points x_0^0, $x_1^0, \ldots,$ x_n^0, so that these points form an orthonormal basis. It means that vectors $\overline{w_1}(0) = x_1^0 - x_0^0$, $\overline{w_2}(0) = x_2^0 - x_0^0 \ldots \overline{w_n}(0) = x_n^0 - x_0^0$ must be orthogonal and have a unit length. Throughout the time these points will move in accordance with the system dynamics to the points $\overline{x_0^1}$, $\overline{x_1^1}, \ldots,$ $\overline{x_n^1}$. We receive vectors $\overline{w_1}(1) = \overline{x_1^1} - \overline{x_0^1}$, $\overline{w_2}(1) = \overline{x_2^1} - \overline{x_0^1}, \ldots,$ $\overline{w_n}(1) = \overline{x_n^1} - \overline{x_0^1}$, which are non-orthogonal and non-unit in a general case. Let us calculate the length of the first vector $d_1(1) = \left|\overline{w_1}(1)\right|$. Now we denote $\overline{x_0^1}$ as x_0^1 and make Gram-Schmidt's orthogonalization:

$$w_1(1) = \frac{\overline{w_1}(1)}{d_1(1)},$$

and then consecutively for $i = \overline{2,n}$ we find vectors

$$u_i(1) = \overline{w_i}(1) - \sum_{j=1}^{i-1}\left(\overline{w_i}(1) \cdot w_j(1)\right) \cdot w_j(1),$$

$$d_i(1) = |u_i(1)|,$$

$$w_i(1) = \frac{u_i(1)}{d_i(1)}.$$

After this procedure, we receive orthonormal vectors $w_1(1)$, $w_2(1), \ldots,$ $w_n(1)$ and projections $d_i(1)$ into them of vectors, which preimages compose on the basis of the previous step. Let us examine the following points $x_1^1 = x_0^1 + w_1(1)$, $x_2^1 = x_0^1 + w_2(1), \ldots,$ $x_n^1 = x_0^1 + w_n(1)$. After the iteration of these points onto the time t we repeat the procedure described. After N number of such steps, we have a set of length measures of every direction $d_i(k)$, $i = \overline{1,n}$, $k = \overline{1,N}$. Now we can estimate Lyapunov's spectrum in accordance with the formula

$$\lambda_i = \frac{1}{N \cdot t} \cdot \sum_{k=1}^{N} \ln d_i(k), \; i = \overline{1,n}.$$

For the exact estimation we have a fairly high value of N.

Using the above method we receive Lyapunov's exponents equal to 0.418 and –1.622 for Henon's system, 0.906, 0 and –14.572 for Lorenz's system, 0.07, 0 and –5.39 for Roessler's system.

9. USE OF NEURAL NETWORK TO ESTIMATE LYAPUNOV'S HIGHEST EXPONENT

Let us consider a forecasting neural network, which allows us to trace the trajectory of any point in a reconstructed phase space. This network after its construction with the appropriate embedding parameter (embedding dimension D and time delay τ) and its approximation property will allow us to make a more exact prediction of a time series. For this purpose, we must have a short series. This network also allows us to reconstruct an attractor from an arbitrary initial point. As a result our network preserves a system dynamics. It means that for every point in the attractor we can take the nearest point, which is far from it at some distance, and then trace its trajectory. In this case, the algorithm of Lyapunov's highest exponent calculation from a small time series can be described in the following way [21–24]:

1. After the construction with the use of the embedding parameter, we train a forecasting neural network using the sliding windows technique.

2. Let us take an arbitrary point $[x(t), x(t + \tau),..., x(t + (D - 2)\tau)]$ in the attractor from the training set and with the use of the multi-step prediction, describe its trajectory $x(t + (D - 1)\tau)$, $x(t + D\tau),...$
3. In the reconstructed phase space, we take the nearest point $[x(t), x(t + \tau),..., x(t + (D - 2)\tau) + d_0]$, where $d_0 \approx 10^{-8}$ and predict its behavior $x'(t + (D - 1)\tau)$, $x'(t + D\tau)$ with the use of a neural network.
4. Let us calculate $\ln d_i = \ln|x'(t + (D - 2 + i)\tau) - x(t + (D - 2 + i)\tau)|$, $i = 1, 2,...$ and take only such points, where we have $\ln d_i < 0$.
5. Plot a graph $\ln d_i$ versus $i\tau$.
6. Let us find a line of regression for taken points and estimate its slope, which is equal to Lyapunov's highest exponent.

According to the above algorithm, we estimate Lyapunov's highest exponents for dynamical systems of Henon's (0.43) and Lorenz's (0.98). In both cases, we used only the *X*-series for the highest exponent calculation. As can be seen, these results are good correlated with those, obtained from a standard algorithm. The advantage of the proposed algorithm is in its simplicity, quickness, accuracy and the necessity of a small amount of data. Figures 13 and 14 present the graphs of $\ln d_i$ versus $i\tau$, and regression lines for the data received from *X*-series of Henon's and Lorenz's systems, respectively.

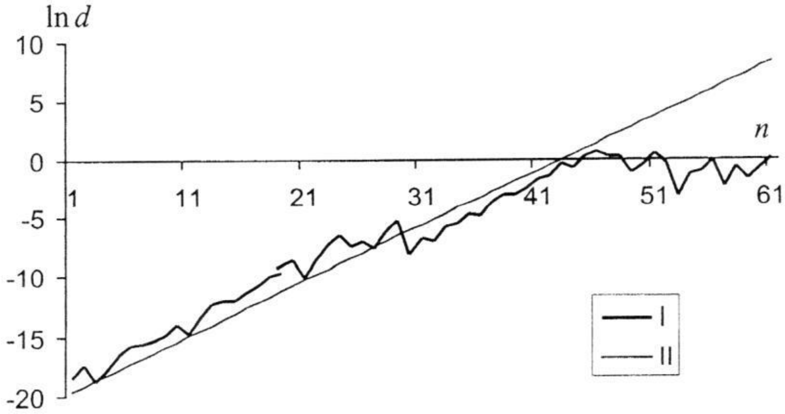

Fig. 13. I – the evolution of the distance between two nearby orbits for Henon's *X*-series; II – the regression line.

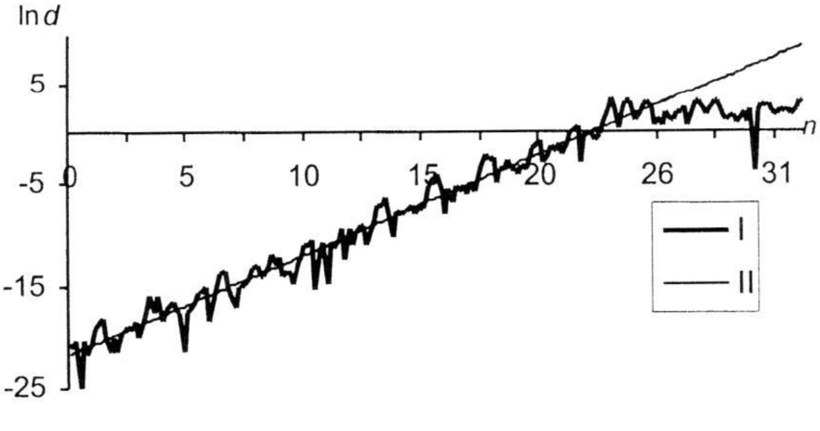

Fig. 14. I – the evolution of the distance between two nearby orbits for Lorenz's *X*-series; II – the regression line.

10. ESTIMATION OF LYAPUNOV'S FULL SPECTRUM WITH USE OF NEURAL NETWORK

A more complicated task is the estimation of Lyapunov's full spectrum from the known time series of all phase variables. The application of Benettin's general algorithm is useless because of the uncertainty of the equations, which describe system dynamics. So we must construct an approximation of the influence of the vector field on the points in the phase space, i.e., for the points $x(t) = (x_1(t), x_2(t), \dots, x_n(t))$, which describe the state of a dynamical system in a time moment t, we must construct mapping $x(t+1) = F(x(t))$ using the given measure of n phase variables. For this purpose, we propose a construction of an approximating two-layer feedforward neural network with nonlinear activation functions in a hidden layer, and linear activation functions in an output layer. The number of neurons in input and output layers is equal to the dimension of phase space n (number of given measuring). In the hidden layer, we have at least $2n+1$ neurons according to Kolmogorov's theorem. Network training is produced based on the following prediction: the input phase points $x(t) = (x_1(t), x_2(t), \dots, x_n(t))$ are associated with the next time $x(t+1) = (x_1(t+1), x_2(t+1), \dots, x_n(t+1))$ in the output layer. A neural network constructed in this way allows us to trace the trajectory of every phase point throughout the time that is not given in advance. We also preserve the system dynamics and measured phase variables of which are used in the network training. After the network training, we can determine the state of a dynamical system in any time moment using only the initial phase point.

We cannot use Benettin's algorithm for this vector field's approximation because the use of an orthonormal basis gives a bad result. So we will use an orthogonal basis with vectors $w_i(t)$, $i = \overline{1,n}$ of small length d on every step. Like this small length we take $d = 10^{-8}$.

Let us examine the algorithm of Lyapunov's full spectrum calculations using the neural network mentioned above [23–27]:

1. Let us take an arbitrary point $y^0(0) = [y_1^0(0), y_2^0(0), \dots, y_n^0(0)]$ in a phase space. We also take points $y^1(0) = [y_1^0(0) + d, y_2^0(0), \dots, y_n^0(0)]$, $y^2(0) = [y_1^0(0), y_2^0(0) + d, \dots, y_n^0(0)], \dots,$ $y^n(0) = [y_1^0(0), y_2^0(0), \dots, y_n^0(0) + d]$, which together with the initial compose an orthogonal basis. In this way we receive basis vectors $w_1(0) = y^1(0) - y^0(0)$, $w_2(0) = y^2(0) - y^0(0), \dots,$ $w_n(0) = y^n(0) - y^0(0)$ with their length equal to $|w_1(0)| = |w_2(0)| = \dots = |w_n(0)| = d$. Let us fix the time $t = 0$.
2. We transfer these points into the input of our network and receive its images $\overline{y^0}(t+1)$, $\overline{y^1}(t+1), \dots,$ $\overline{y^n}(t+1)$, which do not form an orthogonal basis in a general case. We calculate vectors $\overline{w_1}(t+1) = \overline{y^1}(t+1) - \overline{y^0}(t+1)$, $\overline{w_2}(t+1) = \overline{y^2}(t+1) - \overline{y^0}(t+1), \dots,$ $\overline{w_n}(t+1) = \overline{y^n}(t+1) - \overline{y^0}(t+1)$, which are the images of what composes a basis on the previous step.
3. Let us calculate the length $d_1(t+1) = \left|\overline{w_1}(t+1)\right|$. Now we remark $\overline{y^0}(t+1)$ as $y^0(t+1)$ and make Gram-Schmidt's orthogonalization:

$$w_1(t+1) = \frac{\overline{w_1}(t+1)}{d_1(t+1)},$$

and then consecutively for $i = \overline{2,n}$ we find

$$u_i(t+1) = \overline{w_i}(t+1) - \sum_{j=1}^{i-1} \left(\overline{w_i}(t+1) \cdot w_j(t+1)\right) \cdot w_j(t+1),$$

$$d_i(t+1) = |u_i(t+1)|,$$

$$w_i(t+1) = \frac{u_i(t+1)}{d_i(t+1)}.$$

After this step, we receive orthonormal vectors $w_1(t+1)$, $w_2(t+1), \dots,$ $w_n(t+1)$ and projections $d_i(t+1)$ into them of their preimages composing the basis on the previous step. Let us consider the

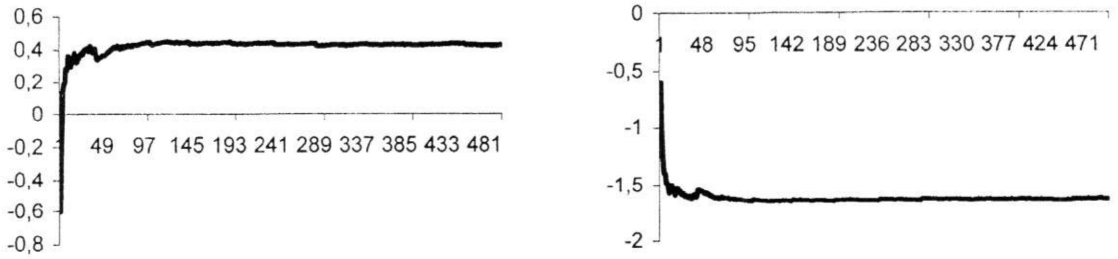

Fig. 15. The estimation of Lyapunov's spectrum for Henon's time series.

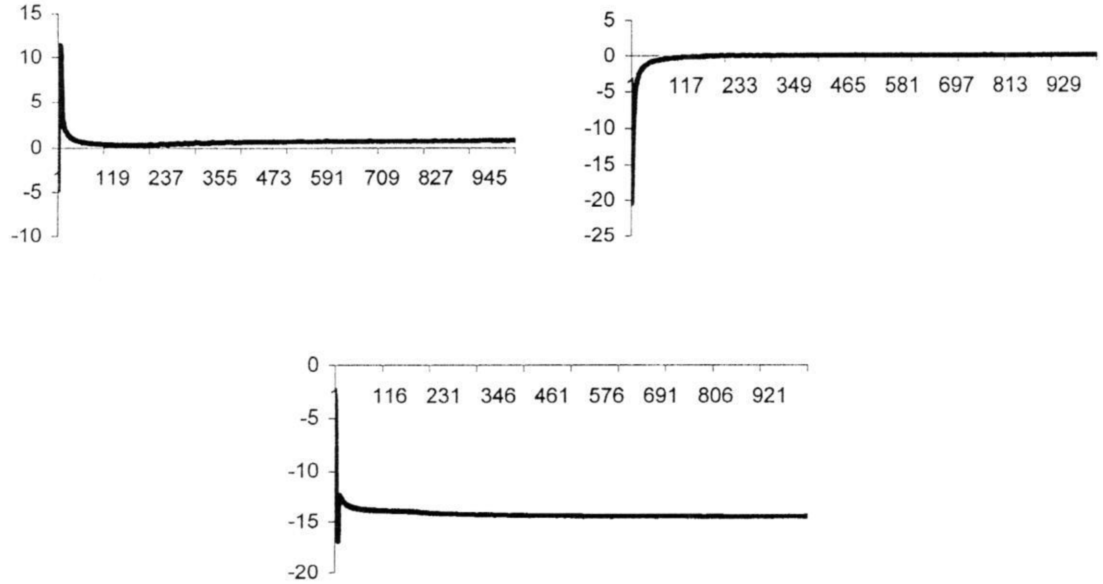

Fig. 16. Estimation of Lyapunov's spectrum for Lorenz's time series.

points $y^1(t + 1) = y^0(t + 1) + d \cdot w_1(t + 1)$, $y^2(t + 1) = y^0(t + 1) + d \cdot w_2(t + 1),\dots, y^n(t + 1) = y^0(t + 1) + d \cdot w_n(t + 1)$, which together with point $y^0(t + 1)$ compose an orthogonal basis on this step.

4. Let us find $s_i(t+1) = \ln\dfrac{d_i(t+1)}{d}$, $i = \overline{1,n}$.

5. Let us give $t := t+1$. If $t < p$, then we go to step 2.

6. We calculate $\lambda_i = \dfrac{1}{p}\sum_{t=1}^{p} s_i(t)$, where $i = \overline{1,n}$, which are the estimation of Lyapunov's sought exponents.

When p increases, the values λ_i become stabilized. So we must take a large value of p ($p \approx 1000$).

Using this method, we calculate Lyapunov's full spectrum for Henon's and Lorenz's systems using the time series of all phase variables.

We obtain Lyapunov's exponents equal to 0.442, –1.625 for Henon's system, and 0.777, 0.003, –14.472 for Lorenz's time series. Figures 15 and 16 present the dependence of λ_i from p for Henon's and Lorenz's time series, respectively.

Table 3. The results of Lyapunov's spectrum estimation for Lorenz's system from only time series using a neural network.

dt	τ	Lyapunov's spectrum			Error	
		λ_1	λ_2	λ_3	Absolute	Relative
0.04170	0.1668	0.612978	–0.2016840	–15.0033	0.559053	3.83%
0.04200	0.1680	0.725777	–0.0211582	–14.6402	0.193839	1.33%
0.04215	0.1686	0.966544	–0.3009800	–15.9458	1.407730	9.64%
0.04220	0.1688	0.965399	–0.3006240	–15.9270	1.389170	9.51%
0.08500	0.1700	1.143851	–0.2816490	–14.9843	0.553092	3.79%
0.04260	0.1704	1.021790	–0.4326160	–15.6514	1.168620	8.00%
0.04260	0.1704	0.483841	0.0528098	–13.3949	1.251610	8.57%
0.04300	0.1720	0.742471	–0.0865899	–14.2650	0.358420	2.45%
0.08600	0.1720	0.830438	–0.3357490	–13.5627	1.066370	7.30%
0.04320	0.1728	0.570654	–0.1465600	–14.9297	0.511766	3.51%
0.08700	0.1740	1.216890	–0.6435080	–14.5374	0.715508	4.90%

Table 4. The results of Lyapunov's spectrum estimation for Roessler's system from only time series using a neural network.

dt	τ	Lyapunov's spectrum			Error	
		λ_1	λ_2	λ_3	Absolute	Relative
0.04	0.04	0.173003	–0.0821049	–5.47571	0.154879	2.87%
0.07	0.07	0.060350	–0.3888620	–5.18352	0.441825	8.19%
0.06	0.12	0.090696	0.0030709	–5.02998	0.363565	6.74%
0.06	0.12	0.106080	–0.0358488	–5.79224	0.402378	7.46%
0.06	0.12	0.077922	–0.0187908	–5.93021	0.537581	9.97%
0.07	0.14	0.129117	–0.1092460	–4.93167	0.477637	8.86%
0.08	0.16	0.106981	–0.0449128	–5.36074	0.065971	1.22%
0.08	0.16	0.085461	–0.0282390	–5.31476	0.084427	1.57%
0.04	0.16	0.119605	–0.2027930	–5.56896	0.272851	5.06%
0.06	0.18	0.141245	–0.0751598	–5.48983	0.141277	2.62%
0.08	0.18	0.078753	–0.0144016	–5.24691	0.147000	2.73%

11. NEURAL NETWORK METHOD OF LYAPUNOV'S FULL SPECTRUM ESTIMATION FROM THE MEASUREMENT OF ONLY ONE VARIABLE

This method uses the same idea as in the case of Lyapunov's highest exponent estimation from only one time series neural networks. The sequence of operations and argumentations are preserved, but we use the neural algorithm, described in the previous section.

Let us revise the order of operations:

1. From the initial time series of one variable we calculate the time delay τ (we can use the following methods: the method based on an autocorrelation function, the method based on a mutual information function or the average displacement method).
2. The calculation of the embedding dimension D (we can use the method of the correlation dimension, the false nearest-neighbors method or a gamma-test).
3. Let's estimate a phase space reconstruction of a trajectory using the time delay method. After this step we receive the sequence of D-dimensional vectors, which size is somewhat smaller than the size of the initial time series.
4. Let us construct an approximate neural network.

5. We train this network to predict the next vector in the reconstructed phase space based on the previous one. The training is successful if an average error of every vector calculation has an order of about 10^{-8}-10^{-9}.
6. Using this neural network, we calculate Lyapunov's full spectrum according to the previous section.

We use this method to calculate Lyapunov's full spectrum for Lorenz's and Roessler's dynamical systems. In every experiment a time series of 400 measurements of an *X*-coordinate series is taken. In the approximate neural network, we use 10 neurons. We take quantization step *dt* and time delay τ as variable parameters. The neural network was trained through 3000 epochs. The tables present the results in which the neural network was trained sufficiently well. As we can see (Tables 3 & 4), the estimation mistake is acceptable in every case. The basic request for the initial time series is an adequacy of the dynamical mapping: the initial time sequence must have points which are distributed uniformly and sufficiently densely in the attractor. However we must note that in some cases (for example in Lorenz's system, Table 3) this method is substantively dependent on the time delay, whereas for Lorenz's system we can take the time delay from a wide range.

REFERENCES

[1] M. Cascadi, "Nonlinear prediction of chaotic time series," *Physica D*, vol. 35, pp. 335-336, 1989.

[2] H. Kantz and T. Schreiber, *Nonlinear time series analysis*, Cambridge: Cambridge University Press, 1997.

[3] K. Juda and M. Small, "Towards long-term prediction," *Physica D*, vol. 136, pp. 31–34, 2000.

[4] J. Eckmann, S. Kamphorst, D. Ruelle, and D. Gillerto, "Lyapunov exponents from a time series," *Phys. Review A*, vol. 34, pp. 4971-4979, 1986.

[5] J. Eckmann and D. Ruelle, "Fundamental limitations for estimating dimensions and Lyapunov exponents in dynamical systems," *Physica D*, vol. 56, pp. 185-187, 1992.

[6] P. Grassberger and I. Procaccia, "Measuring the strangeness of strange attractors," *Physica D*, vol. 9, pp. 189-208, 1983.

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

[8] F. Takens, "Detecting strange attractors in turbulence," Lecture *Notes in Mathematics*, Springer-Verlag, Berlin, vol. 898, pp. 366-381, 1980.

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

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

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

[12] J. Holzfuss and G. Mayer-Kress, "An approach to error estimation in the applications of dimensional algorithms," *Dimensions and Entropies in Chaotic Systems*, editor G. Mayer-Kress, Springer-Verlag, New York, pp. 114-122, 1986.

[13] M.T. Rosenstein, J.J. Colins and C.J. De Luca, "Reconstruction expansion as a geometry-based framework for choosing proper delay time," *Physica D*, vol. 73, pp. 82-98, 1994.

[14] A.M. Fraser and H.L. Swinney, "Independent coordinates for strange attractor from mutual information," *Physical Review A*, vol. 33, pp. 1134-1140, 1986.

[15] 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, no. 4, pp. 1331-1392, 1993.

[16] R. Castro and T. Sauer, "Correlation dimension of attractor through interspike intervals," *Physical Review E*, vol. 55, pp. 287-290, 1997.

[17] M.B. Kennel, R. Brown and H.D.I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Physical Review A*, vol. 45, pp. 3403-3411, 1992.

[18] Stefansson, N. Koncar and A.J. Jones, "A note on the Gamma test," *Neural Computing and Applications*, vol. 5, pp. 387-393, 1997.

[19] G. Benettin, L. Galgani, A. Giorgilli, and 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," *Mechanica,* vol. 15, pp. 9-30, 1980.

[20] A. Wolf, J. Swift, H. Swinney, and J. Vastano, "Determining Lyapunov exponents from a time series," *Physica D*, vol. 16, pp. 285-31, 1985.

[21] V. Golovko, Y. Savitsky and N. Maniakov, "Modeling Nonlinear Dynamics Using Multilayer Neural Networks," Proceedings of the International Workshop on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS'2001, July 1-4, 2001, Foros, Ukraine, Ternopil: Lileya, 2001, pp. 197-202.

[22] 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.

[23] V. Golovko, "From Neural Networks to Intelligent Systems: Selected Aspects of Training, Application and Evolution," chapter of NATO book *Limitations and Future Trends in Neural Computation*, Amsterdam: IOS Press, pp. 219–243, 2003.

[24] V. Golovko, Y. Savitsky and N. Maniakov, "Neural Networks for Signal Processing in Measurement Analysis and Industrial Applications: the Case of Chaotic Signal Processing," chapter of NATO book *Neural networks for instrumentation, measurement and related industrial applications*, Amsterdam: IOS Press, pp. 119-143, 2003.

[25] V. Golovko, "Estimation of Lyapunov's spectrum from one-dimensional observations using neural networks, Proceedings of the second IEEE int. Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS'2003, Lviv, September 8-10, 2003, Piscataway: IEEE Service Center, pp. 95-98, 2003.

[26] V. Golovko and Y. Savitsky, "Computing of Lyapunov's exponents using neural networks," Proceedings of the third international conference on neural networks and artificial intelligence, ICNNAI'2003, Minsk, November 12-14, 2003, Minsk: BSUIR.

[27] V. Golovko and Y. Savitsky, "Computing of Lyapunov's exponents using neural networks," *Computing*, vol. 3, no. 1, pp. 93-98, 2004.