JOURNAL OF OPTIMIZATION, DIFFERENTIAL EQUATIONS AND THEIR APPLICATIONS (JODEA)
Volume **29**, Issue **2**, December **2021**, pp. 76–91, DOI 10.15421/142109

ISSN (print) 2617–0108
ISSN (on-line) 2663–6824

ON THE EQUIVALENCE OF REAL DYNAMIC PROCESS AND ITS NEURAL NETWORK QUADRATIC MODELS

Vasiliy Ye. Belozyorov*, Danylo V. Dantsev†, Yevhen V. Koshel‡

Abstract. A dynamic process given by its own time series $x_1(t) = x(t); t = t_0, t_1, t_2, ...,$ is considered. Using the methods of Recurrence Quantification Analysis (RQA), the dimension n of the embedding space and the optimal time delay τ of the mentioned time series are determined. With the help of these characteristics, the hidden variables $x_2(t) = x(t+\tau), ..., x_n(t) = x(t+(n-1)\tau)$ are restored. Further, the resulting dynamic system $(x_1(t), ..., x_n(t))$ is modeled by a neural network with a quadratic activation function. The simulation result is presented in the form of a system of neural ODEs. After that, the Lyapunov exponents $\lambda_{real,1}, ..., \lambda_{real,n}$ of the real dynamic system and its neural network model $\lambda_{mod,1}, ..., \lambda_{mod,n}$ ordered in the same way are calculated. Then the coincidence of the signs of elements of sequences $\lambda_{real,1}, ..., \lambda_{real,n}$ and $\lambda_{mod,1}, ..., \lambda_{mod,n}$ makes it possible to judge the adequacy (equivalence) of the real dynamic process and its model. Examples are given.

Key words: system of ordinary autonomous differential equations, quadratic activation function, neural network, Lyapunov exponents.

2010 Mathematics Subject Classification: 13A50, 14L24, 15A72, 93C05.

Communicated by Prof. V. Ye. Kapustyan

1. Introduction

One of the fundamental goals of machine learning is modeling and understanding real-world phenomena from observations. Let

$$x_0 = x(t_0), x_1 = x(t_1), ..., x_N = x(t_N) \tag{1.1}$$

be a finite sequence of numerical values of some scalar dynamical variable $x(t)$ measured with the constant time step Δt in the moments $t_i = t_0 + i\Delta t$; $x_i = x(t_i)$; $i = 0, 1, ..., N$. Sequence (1.1) is called a time series [1, 3, 4, 13, 16].

In the future, time series (1.1) will describe any dynamic process obtained from the observation of any one variable characterizing this process, which is measured at regular intervals.

A common practice in chaotic time series analysis has been to reconstruct the phase space by utilizing the delay-coordinate embedding technique, and then

*Department of Applied Mathematics, Oles Honchar Dnipro National University, 72, Gagarin's Avenue, 49010, Dnipro, Ukraine, belozvye2017@gmail.com

†Chaotic Dynamics Incorporated, 201-45793, Luckakuck Way, Chilliwack BC V2R 5S3, Canada, danylo@chaodyna.com

‡Department of Applied Mathematics, Oles Honchar Dnipro National University, 72, Gagarin's Avenue, 49010, Dnipro, Ukraine, eugenefade@gmail.com

to compute the dynamical invariant magnitudes such as unstable periodic orbits, a fractal dimension of the underlying chaotic set, and its Lyapunov spectrum. As a large body of literature exists on applying of the technique of the time series to study chaotic attractors [3, 4, 9, 12, 13, 16], a relatively unexplored issue is its applicability to dynamical systems of differential equations depending on parameters. Our focus will be concentrated on the analysis of influence of parameters of found dynamic system on the behavior of its solutions. These parameters are determined by the structure of the time series (1.1) and choice of approximating functions in right sides of the got system of differential equations.

Let

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t)), \mathbf{x}(t) \in \mathbb{R}^n, \mathbf{F}(\mathbf{x}) \in \mathbb{R}^n, t \in \mathbb{R} \tag{1.2}$$

be a system of ordinary autonomous differential equations and let $\mathbf{x}(t, \mathbf{x}_0)$ be a trajectory of this system with initial data $\mathbf{x}_0 \in \mathbb{R}^n$. Here $\mathbf{F}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n$ is a continuous vector-function; $\mathbf{x}(0, \mathbf{x}_0) = \mathbf{x}_0$.

Let also

$$\dot{\mathbf{y}}(t) = \mathbf{H}(\mathbf{y}(t)), \mathbf{y}(t) \in \mathbb{R}^n, \mathbf{H}(\mathbf{y}) \in \mathbb{R}^n, t \in \mathbb{R} \tag{1.3}$$

be another system of differential equations with initial data $\mathbf{y}_0 \in \mathbb{R}^n$.

In the future, we will assume that point $\mathbf{x} = \mathbf{y} = 0$ is the equilibrium point of systems (1.2) and (1.3) .

Definition 1.1. (see [14]). Two systems of differential equations (1.2) and (1.3) are called topologically equivalent in the neighborhood of equilibrium point 0, if there is a homeomorphism (one-to-one correspondence and mutually continuous mapping) that transfers the equilibrium point $\mathbf{x} = 0$ of system (1.2) and the trajectory lying in some neighborhood of this point, to the equilibrium point $\mathbf{y} = 0$ and the trajectory of system (1.3) lying in some neighborhood of point $\mathbf{y} = 0$ while maintaining the orientation of the trajectories.

Obviously, the question of the equivalence of systems (1.2) and (1.3) can be solved only for the simplest systems (for example, linear). In the case of linear systems, the equivalence at the hyperbolic equilibrium point (0) takes place when the eigenvalues of the Jacobi matrices $\partial\mathbf{F}(\mathbf{x})/\partial\mathbf{x}|_{(0)}$ and $\partial\mathbf{H}(\mathbf{y})/\partial\mathbf{y}|_{(0)}$ coincide [14]. In the case of nonlinear systems, the necessary equivalence condition is the coincidence of the Lyapunov exponents for systems (1.2) and (1.3). This idea is the basis of this work. In its implementation, the Lyapunov exponents are calculated for a real dynamic system and its model, represented as a system of neural ODEs (see [1]).

Consider the time series (1.1). Using the recurrent analysis, we calculate the dimension n of the embedding space and the optimal time delay τ. Using these

characteristics, we construct n time series

$$\begin{cases} x_1(t) = \{x_1(t_0) = x(t_0), x_1(t_1) = x(t_1), ..., x_1(t_m) = x(t_m)\} \\ x_2(t) = \{x_2(t_0) = x(t_0+\tau), x_2(t_1) = x(t_1+\tau), ..., x_2(t_m) = x(t_m+\tau)\} \\ \cdot \\ x_n(t) = \{x_n(t_0) = x(t_0+(n-1)\tau), x_n(t_1) = x(t_1+(n-1)\tau), ..., \\ \qquad x_n(t_m) = x(t_m+(n-1)\tau)\}, \end{cases} \tag{1.4}$$

defining the behavior of a real dynamical system. (Here $t_m + (n-1)\tau < t_N$.)

Using any approximation method (least squares or backward propagation of the error), we construct system (1.2).

Let us denote by $\lambda_{real,1}, ..., \lambda_{real,n}$ (by $\lambda_{mod,1}, \ldots, \lambda_{mod,n}$) the Lyapunov exponents of system (1.4) (system (1.2)). The numbers $\lambda_{real,1}, \ldots, \lambda_{real,n}$ and $\lambda_{mod,1}, \ldots, \lambda_{mod,n}$ are ordered in the same way.

Definition 1.2. Let $\epsilon > 0$ be a given number. Assume that:

(a1) systems (1.2) and (1.4) are dissipative: $\lambda_{mod,1} + ... + \lambda_{mod,n} < 0$ and $\lambda_{real,1} + ... + \lambda_{real,n} < 0$;

(a2) $\forall i \in \{1, ..., n\}$ $\lambda_{real,i} \cdot \lambda_{mod,i} > 0$ and $|\lambda_{real,i} - \lambda_{mod,i}| < \epsilon$.

Then systems (1.2) and (1.4) are called ϵ-equivalent.

2. Designing of the right side of system (1.2)

A choice of equations of model of describing the dynamics of one or another processes is a difficult task. Experiments show that the most logical approach to describing processes in mechanical, hydrodynamic and electrical models is based on the use of well-known physical laws. In particular, these can be the laws of conservation of energy.

By $U(t)$ and $I(t)$ denote respectively the voltage and current in contact electric network. The following laws are most known: the electric energy is accumulated in a capacitor according to the law $E_C = k_C U^2$; the electric energy is accumulated in an inductor according to the law $E_L = k_L I^2$; the electric energy is transformed into heat energy on a resistor according to the law $E_R = k_R UI$. (Here k_C, k_L, and k_R are constants.) In addition, a rate of change of energy $\dot{E}_C = 2k_C U\dot{U}$, $\dot{E}_L = 2k_L I\dot{I}$ or $\dot{E}_R = k_R(\dot{U}I + U\dot{I})$, and magnitudes $U, \dot{U}, I, \dot{I}$ also influences the dynamics of electric network. (It should also be said that the law of conservation of energy in mechanics is based on formula $E = mv^2/2$, where m and v are the mass and velocity of a moving body.)

Thus, a vast class of electrical and mechanical phenomena can be described by quadratic differential equations depending on the linear $U, \dot{U}, I, \dot{I}, v, \dot{v}$, and quadratic U^2, $U\dot{U}$, I^2, $I\dot{I}, UI$, $\dot{U}I, U\dot{I}, v^2, v\dot{v}$ terms.

Let $A = (a_{ij}), B_1, ..., B_n \in \mathbb{R}^{n\times n}$ be real matrices and $c_1, ..., c_n$ are real numbers. In addition, let also the matrices $B_1 = (b_{ij}^{(1)}), ..., B_n = (b_{ij}^{(n)})$ be symmetrical; $i, j = 1, ..., n$.

By $\mathbf{x} = (x_1, ..., x_n)^T$ denote an arbitrary vector of $\mathbb{R}^n$. Consider the system of ordinary autonomous quadratic differential equations

$$\begin{cases} \dot{x}_1(t) = \sum\limits_{j=1}^{n} a_{1j}x_j(t) + \mathbf{x}^T(t)B_1\mathbf{x}(t) + c_1 \equiv F_1(\mathbf{x}(t)), \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ , \\ \dot{x}_n(t) = \sum\limits_{j=1}^{n} a_{nj}x_j(t) + \mathbf{x}^T(t)B_n\mathbf{x}(t) + c_n \equiv F_n(\mathbf{x}(t)) \end{cases} \tag{2.1}$$

of order n with the vector of initial values $\mathbf{x}^T(0) = (x_{10}, ..., x_{n0})$.

All real dynamic processes taking place in the world around us are bounded. Therefore, the future model should be constructed so that for certain values of the parameters, it generates a compact region of attraction.Then the trajectories starting from this region will be attracted to some attractor [11].

It is obvious that the ellipsoid

$$\mathbb{S} \equiv l_1(x_1 + \gamma_1)^2 + ... + l_n(x_n + \gamma_n)^2 - R^2 \leq 0 \tag{2.2}$$

can be taken as the boundary of the region of attraction. (Here $l_1 > 0, ..., l_n > 0, R > 0$.) Then the trajectory starting from this ellipsoid will not go beyond its boundary if condition

$$\dot{\mathbb{S}}_t \equiv 2l_1(x_1(t) + \gamma_1)\dot{x}_1(t) + ... + 2l_n(x_n(t) + \gamma_n)\dot{x}_n(t) \leq 0 \tag{2.3}$$

is satisfied [11].

By virtue of equations (2.1), in order that condition (2.3) to be valid, it is necessary that

$$(l_1x_1, ..., l_nx_n) \cdot \begin{pmatrix} \mathbf{x}^TB_1\mathbf{x} \\ \vdots \\ \mathbf{x}^TB_n\mathbf{x} \end{pmatrix} = l_1x_1\mathbf{x}^TB_1\mathbf{x} + ... + l_nx_n\mathbf{x}^TB_n\mathbf{x} = 0. \tag{2.4}$$

Now we have to find a nonzero vector $\mathbf{l} = (l_1, ..., l_n) \in \mathbb{R}^n$ such that equality (2.4) holds identically with respect to $\mathbf{x}$.

With the help of variable changes, it is always possible to achieve that the ellipsoid (2.2) transforms into a sphere

$$\mathbb{S} \equiv (x_1 + \gamma_1)^2 + ... + (x_n + \gamma_n)^2 - R^2 \leq 0.$$

Therefore, in what follows we can assume that $l_1 = ... = l_n = 1$.

Let $n = 3$. Introduce the designations: $\mathbf{x} = (x, y, z)^T$ and

$$\mathbf{x}^TB_1\mathbf{x} = d_{11}x^2 + d_{12}xy + d_{22}y^2 + d_{13}xz + d_{23}yz + d_{33}z^2,$$

$$\mathbf{x}^TB_2\mathbf{x} = b_{11}x^2 + b_{12}xy + b_{22}y^2 + b_{13}xz + b_{23}yz + b_{33}z^2,$$

$$\mathbf{x}^T B_3 \mathbf{x} = c_{11}x^2 + c_{12}xy + c_{22}y^2 + c_{13}xz + c_{23}yz + c_{33}z^2.$$

Then equation (2.4) can be presented in the form:

$$\begin{gathered} x(d_{11}x^2 + d_{12}xy + d_{22}y^2 + d_{13}xz + d_{23}yz + d_{33}z^2) \\ +y(b_{11}x^2 + b_{12}xy + b_{22}y^2 + b_{13}xz + b_{23}yz + b_{33}z^2) \\ +z(c_{11}x^2 + c_{12}xy + c_{22}y^2 + c_{13}xz + c_{23}yz + c_{33}z^2) \equiv 0. \end{gathered} \tag{2.5}$$

From here it follows that there are only 10 restrictions on the coefficients of the quadatic part of system (2.1):

$$\begin{cases} d_{11} = b_{22} = c_{33} = 0 \\ d_{22} + b_{12} = 0 \\ d_{12} + b_{11} = 0 \\ d_{33} + c_{13} = 0 \\ d_{13} + c_{11} = 0 \\ b_{33} + c_{23} = 0 \\ b_{23} + c_{22} = 0 \\ d_{23} + b_{13} + c_{12} = 0. \end{cases} \tag{2.6}$$

Taking into account the restrictions (2.6), the quadratic part of system (2.1) will depend on 8 parameters p, s, q, d, h, e, f, r. The final form of system (2.1) will be as follows:

$$\begin{cases} \dot{x}(t) = c_1 + a_{11}x + a_{12}y + a_{13}z + pxy + sy^2 + qxz + dyz + hz^2, \\ \dot{y}(t) = c_2 + a_{21}x + a_{22}y + a_{23}z - px^2 - sxy + exz + fyz + rz^2, \\ \dot{z}(t) = c_3 + a_{31}x + a_{32}y + a_{33}z - qx^2 - (d+e)xy - fy^2 - hxz - ryz. \end{cases} \tag{2.7}$$

Consequently, for $n = 3$ it was possible to reduce the number of parameters from 30 to 20.

Now for the existence of the region of attraction, it is necessary to satisfy the inequality (2.3) for $n = 3$ and $l_1 = l_2 = l_3 = 1$. The indicated inequality will be valid if the symmetric matrix

$$H(\gamma_1, \gamma_2, \gamma_3) = \begin{pmatrix} a_{11} & \frac{a_{12}+a_{21}}{2} & \frac{a_{13}+a_{31}}{2} \\ \frac{a_{12}+a_{21}}{2} & a_{22} & \frac{a_{23}+a_{32}}{2} \\ \frac{a_{13}+a_{31}}{2} & \frac{a_{23}+a_{32}}{2} & a_{33} \end{pmatrix} + \gamma_1 \begin{pmatrix} 0 & \frac{p}{2} & \frac{q}{2} \\ \frac{p}{2} & s & \frac{d}{2} \\ \frac{q}{2} & \frac{d}{2} & h \end{pmatrix}$$

$$+\gamma_2 \begin{pmatrix} -p & -\frac{s}{2} & \frac{e}{2} \\ -\frac{s}{2} & 0 & \frac{f}{2} \\ \frac{e}{2} & \frac{f}{2} & r \end{pmatrix} + \gamma_3 \begin{pmatrix} -q & -\frac{d+e}{2} & -\frac{h}{2} \\ -\frac{d+e}{2} & -f & -\frac{r}{2} \\ -\frac{h}{2} & -\frac{r}{2} & 0 \end{pmatrix}$$

is negative definite.

The last restriction can be achieved if for some real $\gamma_1, \gamma_2, \gamma_3$ the conditions of Sylvester (see [8, 11])

$$\begin{cases} \Delta_1(\gamma_1,\gamma_2,\gamma_3) = a_{11} - p\gamma_2 - q\gamma_3 < 0, \\ \Delta_2(\gamma_1,\gamma_2,\gamma_3) = (a_{11} - p\gamma_2 - q\gamma_3)(a_{22} + s\gamma_1 - f\gamma_3) \\ \qquad\qquad - \dfrac{(a_{12} + a_{21} + p\gamma_1 - s\gamma_2 - (d+e)\gamma_3)^2}{4} > 0, \\ \Delta_3(\gamma_1,\gamma_2,\gamma_3) = \det H(\gamma_1,\gamma_2,\gamma_3) < 0 \end{cases} \tag{2.8}$$

are true.

2.1. Generalization to n-dimensional case

Let us again use the relation (2.4), in which it is assumed that $l_1 = ... = l_n = 1$. As a result, we will have three groups of restrictions:
C_n^1 one-term restrictions

$$b_{ii}^{(i)} x_i^3 \equiv 0; i = 1, ..., n; \tag{2.9}$$

$2C_n^2$ two-term restrictions

$$b_{jj}^{(i)} x_i x_j^2 + b_{ij}^{(j)} x_i x_j^2 \equiv 0; i \neq j; i, j = 1, ..., n; \tag{2.10}$$

C_n^3 three-term restrictions

$$b_{jk}^{(i)} x_i x_j x_k + b_{ik}^{(j)} x_i x_j x_k + b_{ij}^{(k)} x_i x_j x_k \equiv 0; i \neq j \neq k; i, j, k = 1, ..., n. \tag{2.11}$$

Thus, from (2.9), (2.10), and (2.11) it follows that in the quadratic part of system (2.1), only

$$\phi(n) = nC_{n+1}^2 - C_n^1 - 2C_n^2 - C_n^3 = \frac{n(n^2-1)}{3}$$

coefficients will be independent. For example, for $n = 2, 3$, and 4, we have $\phi(2) = 2, \phi(3) = 8$, and $\phi(4) = 20$.

Taking into account the restrictions (2.9) – (2.11), the quadratic part of system system (2.1) will depend on 20 parameters at $n = 4$ (instead of 40 parameters). The final form of system (2.1) (at $n = 4$) will be as follows:

$$\begin{cases} \dot{x}(t) = c_1 + a_{11}x + a_{12}y + a_{13}z + a_{14}u + b_{12}xy + b_{13}xz + b_{14}xu + b_{22}y^2 \\ \qquad +b_{23}yz + b_{24}yu + b_{33}z^2 + b_{34}zu + b_{44}u^2, \\ \dot{y}(t) = c_2 + a_{21}x + a_{22}y + a_{23}z + a_{24}u - b_{12}x^2 - b_{22}xy + c_{13}xz + c_{14}xu \\ \qquad +c_{23}yz + c_{24}yu + c_{33}z^2 + c_{34}zu + c_{44}u^2, \\ \dot{z}(t) = c_3 + a_{31}x + a_{32}y + a_{33}z + a_{34}u - b_{13}x^2 - (b_{23} + c_{13})xy - b_{33}xz \\ \qquad +d_{14}xu - c_{23}y^2 - c_{33}yz + d_{24}yu + d_{34}zu + d_{44}u^2, \\ \dot{u}(t) = c_4 + a_{41}x + a_{42}y + a_{43}z + a_{44}u - b_{14}x^2 - (b_{24} + c_{14})xy \\ \qquad -(b_{34} + d_{14})xz - b_{44}xu - c_{24}y^2 - (c_{34} + d_{24})yz - c_{44}yu - d_{34}z^2 - d_{44}zu \end{cases} \tag{2.12}$$

It remains to show how it is possible to transform system (2.1) into a system of neural ODEs. To do this, it is necessary to introduce a new variable x_0 in the following way: instead of constants c_i, introduce the terms $c_i x_0^2$, and instead of the linear terms $a_{ij}x_j$, introduce the quadratic terms $a_{ij}x_j x_0; j \neq 0$. As a result, system (2.1) will turn into a system of differential equations whose right-hand sides are quadratic forms in $n+1$ variables $x_0, x_1, ..., x_n$. Now applying to each of these quadratic forms the method of reduction to the sum of squares (for example, the Lagrange method [8]) and after that setting $x_0 = 1$, we represent the system (2.1) in the following aspect

$$\begin{cases} \dot{x}_1(t) = \alpha_{11} \cdot \Big(\sum\limits_{j=1}^{n} \beta_{1j}^{(1)} x_j + \gamma_1^{(1)}\Big)^2 + ... + \alpha_{1,n+1} \cdot \Big(\sum\limits_{j=1}^{n} \beta_{n+1,j}^{(1)} x_j + \gamma_{n+1}^{(1)}\Big)^2 \\ . \ , \\ \dot{x}_n(t) = \alpha_{n1} \cdot \Big(\sum\limits_{j=1}^{n} \beta_{1j}^{(n)} x_j + \gamma_1^{(n)}\Big)^2 + ... + \alpha_{n,n+1} \cdot \Big(\sum\limits_{j=1}^{n} \beta_{n+1,j}^{(n)} x_j + \gamma_{n+1}^{(n)}\Big)^2. \end{cases} \tag{2.13}$$

This is the system of neural quadratic ODEs with the quadratic activation function $\psi(x) = x^2$ [1,2,6,9].

2.2. Algorithm for constructing model (2.1) from known time series (1.4)

Suppose that n time series

$$\begin{gathered} g_{10}, g_{11}, g_{12}, ..., g_{1N} \\ g_{20}, g_{21}, g_{22}, ..., g_{2N} \\ ..... \\ g_{n0}, g_{n1}, g_{n2}, ..., g_{nN} \end{gathered}$$

are given on the same time interval T in equally spaced N nodes: $0, \Delta t, ..., k\Delta t$, ..., $N\Delta t = T$. Thus, $\Delta t = T/N$.

Assume that $n = 3$. The model describing the process should be presented in the form of system (2.7). (Model (2.7) weights will be determined using the least squares method [9].)

1. Fix a learning selections

$$g_{10}, g_{11}, g_{12}, ..., g_{1m};\ g_{20}, g_{21}, g_{22}, ..., g_{2m};\ g_{30}, g_{31}, g_{32}, ..., g_{3m},$$

   where $0 < 20 \leq m \leq N$.

2. Let $g_1 = x, g_2 = y, g_3 = z$. From here it follows that $g_{1j} = x(t_j) = x_j, g_{2j} = y(t_j) = y_j, g_{3j} = z(t_j) = z_j; j = 1, ..., m$.

3. Construct the column of numerical derivatives:

$$D = \begin{pmatrix} D_1 \\ D_2 \\ D_3 \end{pmatrix} \in \mathbb{R}^{3m}, \text{where } D_i = \frac{1}{\Delta t} \begin{pmatrix} g_{i1} - g_{i0} \\ g_{i2} - g_{i1} \\ \vdots \\ g_{i,m} - g_{i,m-1} \end{pmatrix} \in \mathbb{R}^m; i = 1, ..., 3.$$

4. Construct the matrix of Jacobi of sizes $(3m \times 20)$:

$$J = \left(\begin{array}{cccccccccccc|c} 1 & x_1 & y_1 & z_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 1 & x_2 & y_2 & z_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ 1 & x_m & y_m & z_m & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ - & - & - & - & - & - & - & - & - & - & - & - & \\ 0 & 0 & 0 & 0 & 1 & x_1 & y_1 & z_1 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 1 & x_2 & y_2 & z_2 & 0 & 0 & 0 & 0 & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & 0 & 1 & x_m & y_m & z_m & 0 & 0 & 0 & 0 & \\ - & - & - & - & - & - & - & - & - & - & - & - & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_1 & y_1 & z_1 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_2 & y_2 & z_2 & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_m & y_m & z_m & \end{array} \right| \rightarrow$$

$$\left| \begin{array}{cccccccc} x_1y_1 & y_1^2 & x_1z_1 & y_1z_1 & z_1^2 & 0 & 0 & 0 \\ x_2y_2 & y_2^2 & x_2z_2 & y_2z_2 & z_2^2 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_my_m & y_m^2 & x_mz_m & y_mz_m & z_m^2 & 0 & 0 & 0 \\ -- & -- & -- & -- & -- & -- & -- & -- \\ -x_1^2 & -x_1y_1 & 0 & 0 & 0 & x_1z_1 & y_1z_1 & z_1^2 \\ -x_2^2 & -x_2y_2 & 0 & 0 & 0 & x_2z_2 & y_2z_2 & z_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -x_m^2 & -x_my_m & 0 & 0 & 0 & x_mz_m & y_mz_m & z_m^2 \\ -- & -- & -- & -- & -- & -- & -- & -- \\ 0 & 0 & -x_1^2 & -x_1y_1 & -x_1z_1 & -x_1y_1 & -y_1^2 & -y_1z_1 \\ 0 & 0 & -x_2^2 & -x_2y_2 & -x_2z_2 & -x_2y_2 & -y_2^2 & -y_2z_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & -x_m^2 & -x_my_m & -x_mz_m & -x_my_m & -y_m^2 & -y_mz_m \end{array} \right)$$

5. Perform replacements $x_j = g_{1j}, y_j = g_{2j}, z_j = g_{3j}; j = 1, ..., m$, and check the condition: $\operatorname{rank} J = 20$. If $\operatorname{rank} J < 20$ then fix some number $\mu \in (0, 1]$ else $\mu := 0$.

6. Put Y^T

$$= (c_1, a_{11}, a_{12}, a_{13}, c_2, a_{21}, a_{22}, a_{23}, c_3, a_{31}, a_{32}, a_{33}, p, s, q, d, h, e, f, r) \in \mathbb{R}^{20}$$

and compute the vector

$$Y = (J^T \cdot J + \mu I)^{-1} \cdot J^T \cdot D \in \mathbb{R}^{20},$$

where $I \in \mathbb{R}^{20\times 20}$ is an identity matrix.

7. Check the solvability with respect to unknowns $\gamma_1, \gamma_2, \gamma_3$ of the system of inequalities (2.8).

8. If system (2.8) is solvable, then find parameters $\gamma_1, \gamma_2, \gamma_3$ and solve system (2.7) with the initial conditions $x_0 = -\gamma_1, y_0 = -\gamma_2, z_0 = -\gamma_3$. If solutions of this system are bounded, then stop the algorithm; otherwise, use as initial conditions another solution of the system of inequalities (2.8).

9. If system (2.8) is unsolvable, then change the training sample and go to step 1 of the algorithm.

Comment 1. After the design of system (2.7), its transformation into system (2.13) is not difficult.
Comment 2. Note that the proposed algorithm is much simpler than classical neural network modeling [9, 13]. Indeed, in model (2.13) at $n = 3$, it is necessary to adjust 60 parameters; while in model (2.7) only 20 parameters are adjusted.
Comment 3. If it is not possible to build a model that adequately describes the process under study, it is necessary to reduce the number of weight coefficients.

By $\mathbb{W} = \{c_i, a_{ij}, b_{ij}^{(k)}\}$ denote the parametric space of all weights of system (2.1) (the dimension of this space is $L = n(n + 1)(n + 2)/2$). Obviously, the magnitude of the error of approximation Err of time series (1.4) by system (2.1) calculated using the least squares method depends on the choice of basis in $\mathbb{W}$ (on the number of weights in (2.1)). Therefore, looking through all possible subspaces $\mathbb{W}_k \subset \mathbb{W}$ of dimension $L - k$, it is possible to reduce the magnitude Err; $k \geq 1$.

For example, if we want to remove the k-th parameter from model (2.7), we need to remove the k-th column of the Jacobi matrix; $k \in \{1, ..., 20\}$ (here $L = 20$). In this case, the algorithm works with a Jacobi matrix of sizes $3m \times 19$.

3. Calculation of Lyapunov exponents for time series (1.4)

Computing the Lyapunov exponents is trivial when the system is given by its analytical representation and there exist multiple methods for doing this. However, when the data is experimental and the underlying system is unknown, multiple problems arise. For example, it is not possible to directly use a QR-based algorithm for computation of the Lyapunov spectrum as described in [7] because the Jacobian of the system is not known. Because of that, multiple methods for

estimating the Lyapunov exponents were proposed [10]. One of such methods adapts the QR-based approach and makes it possible to apply it for observed time series.

The QR-method for observed time series was developed and investigated by Brown et al. [4]. It utilized an approach for estimating the local Jacobian matrix from the reconstructed trajectory and used the estimated matrix to proceed with the QR factorization and accumulation of the expansion rates estimates as described in [7]. The algorithm provides satisfactory results for the model examples.

The most widely known algorithm for estimating the Lyapunov exponents spectrum from experimental numerical data is based on accumulating the local expansion/contraction rates using the jacobian matrix properties. Suppose that the system that generates the attractor is described by the equation (1.2). The algorithm is formulated as follows [5, 7, 10]:

1. Pick a set $Y \subset \mathbb{R}^{n\times n}$ of orthogonal vectors in the tangent space of the attractor and initialize the vector $L \in \mathbb{R}^n$ of Lyapunov exponents with zeroes.

2. Pick a point on the attractor.

3. Calculate the local jacobian matrix as

$$DF(x) = I + J(x)dt \in \mathbb{R}^{n\times n}, \tag{3.1}$$

   where I is the identity matrix, $J(x) \in \mathbb{R}^{n\times n}$ is the value of the jacobian at the given point, and dt is the step of integration.

4. Multiply the set Y (from the step 1) by the $DF(x)$ matrix and perform QR decomposition on the resulting matrix.

5. Add the absolute values of the diagonal elements of the R matrix obtained at the step 4 to the vector L (from the step 1).

6. Assign $Y = Q$ and return to the step 2 until there are no points left on the attractor.

7. Scale the vector L by dividing it by Ndt, where N is the total number of processed points and dt is the step of integration.

It gives a good approximation for the true values of the Lyapunov exponents (see the figures below for examples).

However, it requires the explicit jacobian matrix which is not available when estimating the Lyapunov exponents for experimental data. That's why we need some method for estimating the local expansion rate having only the experimental data.

Let $a = (a_1, ..., a_n)^T \in \mathbb{R}^n$. The DF matrix is defined as [11]:

$$x_{k+1} = a + DFx_k \in \mathbb{R}^n; k = 0, 1, ... \tag{3.2}$$

In other words, the DF matrix maps the point where it is calculated to the next point in the data series. In order to estimate this matrix, we can use the following method.

Firstly, let's rewrite the (3.2) in the matrix form:

$$\begin{pmatrix} x_{(k+1)1} \\ \vdots \\ x_{(k+1)n} \end{pmatrix} = \begin{pmatrix} a_1 & DF_{11} & \dots & DF_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_n & DF_{n1} & \dots & DF_{nn} \end{pmatrix} \begin{pmatrix} 1 \\ x_{k1} \\ \vdots \\ x_{kn} \end{pmatrix}, \tag{3.3}$$

here a is a scaling constant, x_{ij} is the j-th element of the i-th point in the dataset, DF_{ij} is the element of the DF matrix.

From (3.3) it is clear that the i-th component of the x_{k+1} vector is given by:

$$x_{(k+1)i} = a_i + \sum_{j=1}^{n} DF_{ij}x_{kj}. \tag{3.4}$$

The equation (3.4) has too many variables to solve for DF_{ij} precisely. We can mitigate this issue by constructing multiple equations and combining them into a linear system that is solved trivially. Other equations in this case will be obtained by selecting at least $n-1$ neighbouring points of x_k and constructing equations of type (3.3) for them. If everything is done correctly, the i-th row of the DF matrix can be found from the following equation:

$$\begin{pmatrix} x_{(k_1+1)i} \\ \vdots \\ x_{(k_S+1)i} \end{pmatrix} = \begin{pmatrix} 1 & x_{k_1 1} & \dots & x_{k_1 n} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{k_S 1} & \dots & x_{k_S n} \end{pmatrix} \begin{pmatrix} a_i \\ DF_{i1} \\ \vdots \\ DF_{in} \end{pmatrix}, \tag{3.5}$$

where $x_{k_s} \in \{x_{k_i} \mid \|x_{k_i} - x_n\| < \epsilon\}$; $s \in [n, \dots, S > n]$; $i \in [1, ..., n]$.

By constructing and solving the (3.5) equation m times (to obtain each row of the DF matrix), we get a complete DF matrix that can be used in the algorithm described earlier. Unfortunately, there is no way of finding the dt value from the equation (3.1) having only experimental data. This is the reason why the exponents found using this algorithm with an estimated DF matrix are going to have wrong scales, but correct signs and relative magnitude. Examples for using this approach are described below.

4. Examples

In this section, the operability of the algorithms presented in subsections 2.2 and 3.1 is checked.

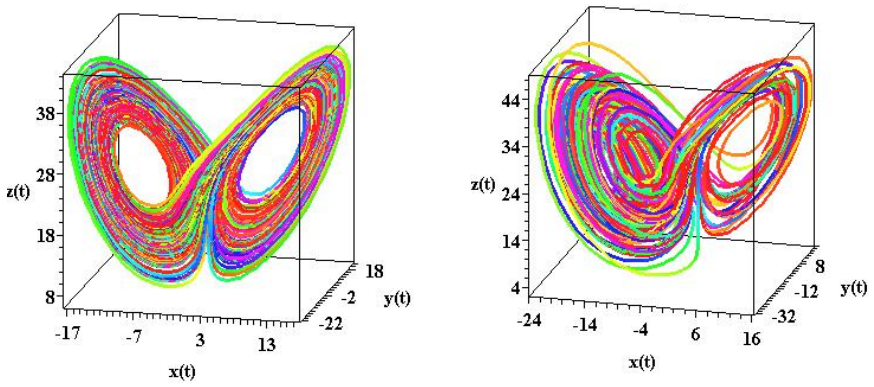
1. Consider the classical Lorenz system [15]

$$\begin{cases} \dot{x}(t) = -10x + 10y, \\ \dot{y}(t) = 28x - y - xz, \\ \dot{z}(t) = -(8/3)z + xy \end{cases} \tag{4.1}$$

and its reconstruction (2.7)

$$\begin{cases} \dot{x}(t) = 2.42 - 10.00x + 9.89y - 0.21z, \\ \dot{y}(t) = -1.55 + 33.11x - 4.36y + 0.27z - 1.17xz + 0.13yz, \\ \dot{z}(t) = -3.04 + 0.18x - 0.24y - 2.50z + 1.17xy - 0.13y^2 \end{cases} \tag{4.2}$$

obtained from one measured variable $x(t)$ using the algorithm of subsection 2.2. Lyapunov exponents for systems (4.1) and (4.2) calculated by the algorithm of section 3 are $(0.77, -0.008, -14.48)$ and $(0.97, -0.04, -14.57)$, respectively. It is clear that these systems are 0.2-equivalent (see Fig.1).



(a1) (a2)

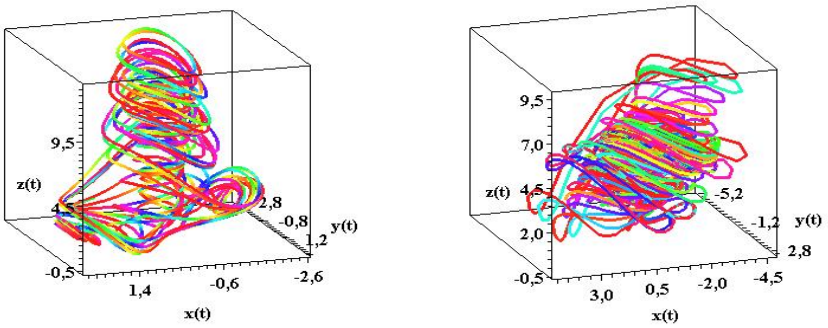
Fig. 4.1. The attractors of systems (4.1) (a1) and (4.2) (a2)

2. Consider Chen-like system [15]

$$\begin{cases} \dot{x}(t) = -0.4x + y, \\ \dot{y}(t) = x + 0.3y - xz, \\ \dot{z}(t) = -0.1z + y^2 - 1. \end{cases} \tag{4.3}$$

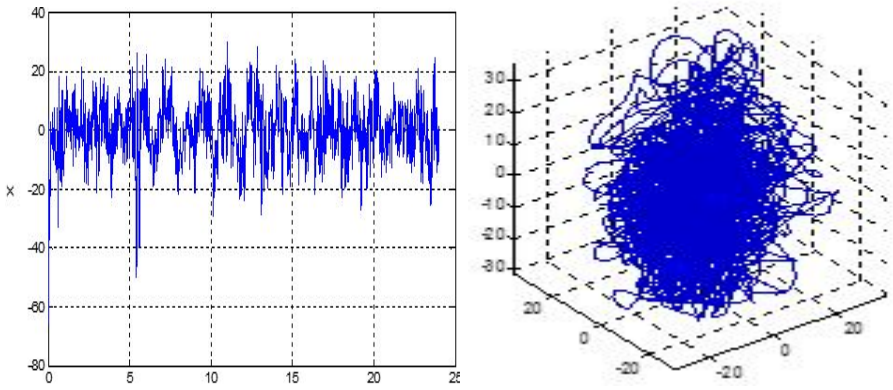
and its reconstruction (2.7)

$$\begin{cases} \dot{x}(t) = 0.9048 - 9.6441x + 38.1911y + 0.1468y^2 - 2.0531xy + 1.0774xz \\ \qquad -0.6151yz, \\ \dot{y}(t) = -1.4317 - 20.6880x + 51.1361y - 0.1625z + 2.0531x^2 - 0.1468xy \\ \qquad -0.2955xz - 12.5954yz, \\ \dot{z}(t) = -9.0557 - 6.2404x + 2.6827y + 0.2446z - 1.0774x^2 + 12.5954y^2 \\ \qquad +0.9106xy \end{cases} \tag{4.4}$$



(a1) (a2)

Fig. 4.2. The attractors of systems (4.3) (a1) and (4.4) (a2)



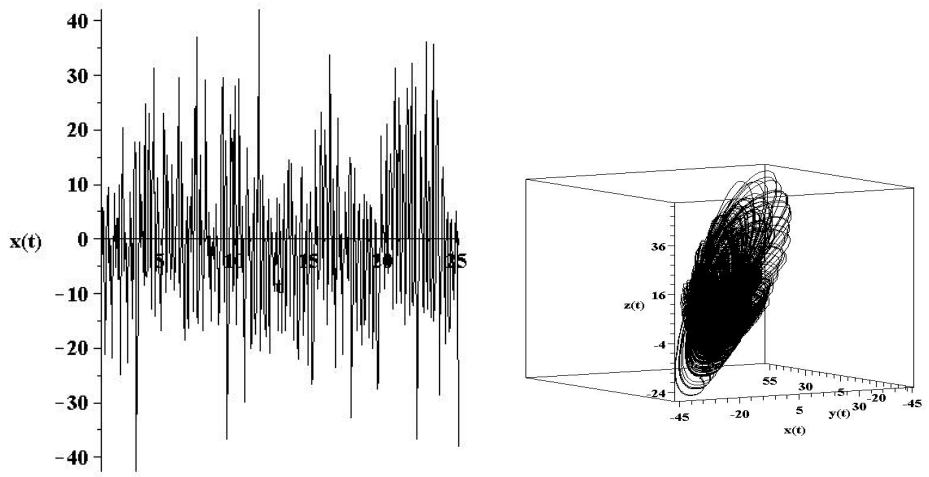
(a1) (a2)

Fig. 4.3. The original signal $x(t)$ at point O1 (a1) and the chaotic attractor restored at this point by variable $x(t)$ (a2)

obtained from one measured variable $x(t)$ using the algorithm of subsection 2.2. Lyapunov exponents for systems (4.3) and (4.4) calculated by the algorithm of section 3 are $(0.21, -0.004, -0.41)$ and $(0.10, -0.02, -0.43)$, respectively. It is clear that these systems are 0.11-equivalent. Nevertheless, it should be noted that the found attractors are not topologically equivalent (see Fig.2).

3. Consider the process of behavior of signal at point O1 of the cerebral cortex of a patient suffering from epilepsy [3]. For this point, the dimension of the embedding space ($n = 3$) and the delay time ($\tau = 34$) were determined [16]. On the basis of these data, using RQA methods [16], an attractor defining the process of disease development was built (see Fig.3).

Now we will use the least squares method in the form presented in [3] to build model (2.1) of the process shown in Fig.3. For this purpose, points from 500 to 2500 of time series Fig.4(a1) were used. In another model, points from 1500 to



(a1) (a2)

Fig. 4.4. The signal $x(t)$ of system (4.5) (a1) and the chaotic attractor generated by this system (a2)

3000 of time series Fig.5(a1) were also used. As a result, we obtained the following systems of differential equations:

$$\begin{cases} \dot{x}(t) = 727 - 34x + 91.6y - 79.8z - 8.5x^2 \\ \qquad -7.9y^2 - 2.5z^2 + 16.5xy + 4.7xz - 3.0yz, \\ \dot{y}(t) = 933.7 - 121.6x + 104.3y + 12.2z - 19.9x^2 \\ \qquad -8.6y^2 - 4.8z^2 + 26.9xy + 17.4xz - 12.1yz, \\ \dot{z}(t) = -189.7 + 121.5x - 47.2y - 63.2z + 3.9x^2 \\ \qquad +3.8y^2 - 0.71z^2 - 7.4xy - 0.74xz + 1.9yz \end{cases} \tag{4.5}$$

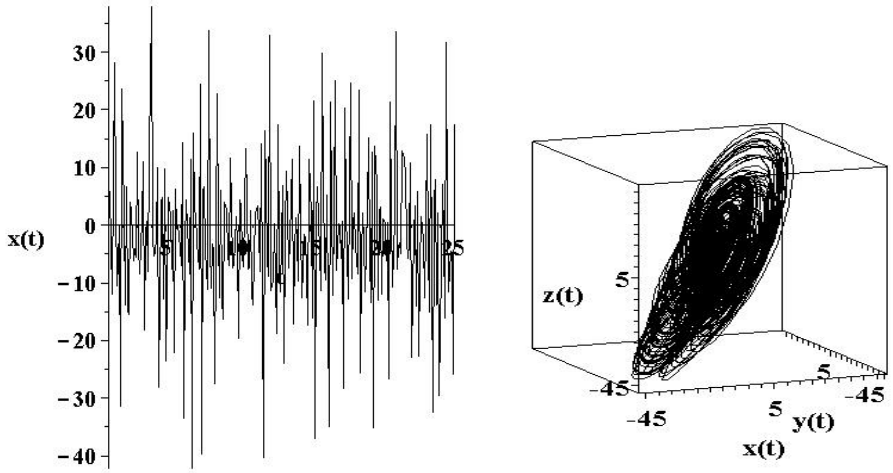
and

$$\begin{cases} \dot{x}(t) = 7 \cdot (28.58 + 12.6x - 11.87y - 2.574z - 0.0231x^2 \\ \qquad -0.2y^2 - 0.130z^2 + 0.11xy - 0.02xz + 0.23yz), \\ \dot{y}(t) = 7 \cdot (36.29 + 15.35x - 11.01y - 3.045z - 0.21x^2 \\ \qquad -0.192y^2 - 0.19z^2 + 0.24xy + 0.29xz + 0.07yz), \\ \dot{z}(t) = 7 \cdot (5.94 + 12.74x - 9.62y - 2.36z + 0.357x^2 \\ \qquad +0.15y^2 - 0.13z^2 - 0.64xy + 0.06xz + 0.23yz). \end{cases} \tag{4.6}$$

Thus, comparing figures Fig.3, Fig.4, and Fig.5, we can say that modeling the real dynamic process of epilepsy using equations (4.5) and (4.6) gives a satisfactory result: Lyapunov exponents are

$$(5.89301, 0.37, -14.4721),\ (6.6868, -0.0007, -11.4912),$$
$$\text{and}\ \ (5.9565, -0.41, -12.1923),$$

respectively. The main reason for this result is that epilepsy is a nonstationary process and new instrumentation will need to be used to describe it. (The method



(a1) (a2)

Fig. 4.5. The signal $x(t)$ of system (4.6) (a1) and the chaotic attractor generated by this system (a2)

proposed in [3] differs from the algorithm of subsection 2.2 only in the construction of the Jacobi matrix.)

5. Conclusion

The article proposes a new algorithm for reconstructing a system of quadratic differential equations from a known time series. The essence of the algorithm is that it guarantees the existence of a spherical region of the phase space, in which the dynamic process generated by the found ODE system is concentrated.

To solve the equivalence problem, a new algorithm for estimating the Lyapunov exponents of a given time series is also proposed. This algorithm correctly estimates the signs of the Lyapunov exponents, but the results suffer from the incorrect scaling. The reason for this is that the QR decomposition algorithm with the analytical representation of the Jacobian uses the integration step value to scale the accumulated local expansion rates at the end of the algorithm. When the estimated Jacobian is used information about the integration step is not available and the results are not scaled properly.

References

1. V. Ye. Belozyorov, D. V. Dantsev, S. A. Volkova, *On equivalence of linear control systems and its usage to verification of the adequacy of different models for a real dynamic process*, Journal of Optimization, Differential Equations and Their Applications (JODEA), **28**(1)(2020), 43–97.
2. V. Ye. Belozyorov, D. V. Dantsev, *Stability of neural ordinary differential equations with power nonlinearities*, Journal of Optimization, Differential Equations and Their Applications (JODEA), **28**(2)(2020), 21–46.

3. V. Ye. Belozyorov, O. V. Pohorielov, V. N. Serdiuk, V. G. Zaytsev, *New approach to problem of diagnostics of cerebral cortex diseases using chaotic dynamics methods*, *In: The collection of 7 th The international conference "Social Science and Humanity"*, September 23 – 29, London, **2**(2017), 7–28.
4. R. Brown, P. Bryant, H. D. I. Abarbanel, *Computing the Lyapunov spectrum of a dynamical system from an observed time series*, *Physical Review A*, **43**(1991), 2787–2806.
5. H. F. Bremen, F. E. Udwadia, W. Proskurowski, *An efficient QR based method for the computation of Lyapunov exponents*, *Physica D: Nonlinear Phenomena*, **101**(1997), 1–16.
6. R. T. Q. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud, *Neural ordinary differential equations*, *arXiv preprint arXiv:1806.07366v5[cs.LG]*, (2019), 1–18.
7. F. Christiansen, H. H. Rugh, *Computing Lyapunov spectra with continuous Gram-Schmidt orthonormalization*, *arXiv:chao-dyn/9611014v1*, (1996), 1–14.
8. F. R. Gantmacher, *The Theory of Matrices*, AMS Chelsea Publising, Providence, Rhode Island, 2000.
9. S. Haykin, *Neural Networks. A Comprehensive Foundation*, Second Edition, Pearson Education, Prentice Hall, 2005.
10. Y. V. Koshel, *Local jacobian estimation for delay embedded time series data*, *In: The abstracts of 18th international scientific and practical conference "MPZIS – 2021"*, November 17 – 19, Dnipro, (2021), 152–153.
11. H. K. Khalil, *Nonlinear Systems – 2nd Edition*, (Prentice Hall/New-Jersy), 1996.
12. Q. Li, P. Xu, *Estimation of Lyapunov spectrum and model selection for a chaotic time series*, *Applied Mathematical Modelling*, **36**(2012), 6090–6099.
13. Z. Liu, *Chaotic time series analysis*, *Mathematical Problems in Engineering*, **2010** (2010), 720190, 1–31.
14. G. Teschi, *Ordinary Differential Equations and Dynamical Systems*, Graduate Studies in Mathematics, **140**, AMS, Providence, Rhode Island, 2012.
15. X. Wang, G. Chen, *A gallery of Lorenz-like and Chen-like attractors*, *International Journal of Bifurcation and Chaos*, **23**(2013), 1330011, 1–20.
16. C. L. Webber, N. Marwan (eds), *Reccurence Quantification Analysis. Theory and Best Practice*, Springer, NY, Dordrecht, Heidelberg, London, 2015.

Received 30.10.2021