Neurocomputing 74 (2011) 1735–1747

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Short-term time series forecasting based on the identification of skeleton algebraic sequences

Minvydas Ragulskis [a,*], Kristina Lukoseviciute [a,1], Zenonas Navickas [b,2], Rita Palivonaite [a,3]

[a] Research Group for Mathematical and Numerical Analysis of Dynamical Systems, Kaunas University of Technology, Studentu 50-325, Kaunas LT-51368, Lithuania
[b] Department of Applied Mathematics, Kaunas University of Technology, Studentu 50-325, Kaunas LT-51368, Lithuania

ARTICLE INFO

Article history:
Received 26 August 2010
Received in revised form
25 November 2010
Accepted 14 February 2011
Communicated by G.P. Zhang
Available online 21 March 2011

Keywords:
Hankel matrix
Time series forecasting
Algebraic sequence
Evolutionary algorithms

ABSTRACT

A new short-term time series forecasting method based on the identification of skeleton algebraic sequences is proposed in this paper. The concept of the rank of the Hankel matrix is exploited to detect a base fragment of the time series. Particle swarm optimization and evolutionary algorithms are then used to remove the noise and identify the skeleton algebraic sequence. Numerical experiments with an artificially generated and a real-world time series are used to illustrate the functionality of the proposed method.

1. Introduction

Time series prediction, especially short time series prediction, is a challenging problem in many fields of science and engineering. Many techniques exist for time series forecasting. In general, the objective of these techniques is to build a model of the process and then use this model on the last values of the time series to extrapolate past behavior into future. Forecasting procedures include different techniques and models. The use of general exponential smoothing to develop an adaptive short-term forecasting system based on observed values of integrated hourly demand is explored in [1]. A self-organizing fuzzy neural network is used to predict a real-time short-peak and average load in [2]. Day-ahead prices are forecasted with genetic-algorithm-optimized Support Vector Machines in [3]. Artificial neural networks (ANN) are used for day-ahead price forecasting in restructured power systems in [4]. A fuzzy logic based inference system is used for the prediction of short-term power prices in [5]. A hybrid approach based on ANN and genetic algorithms is used for detecting temporal patterns in stock markets in [6]. Short-time traffic flow prediction based on chaotic time series theory is developed in [7]. The radial basis function ANN with a nonlinear time-varying evolution particle swarm optimization (PSO) algorithm is used to forecast one-day ahead and five-days ahead of a practical power system in [8]. PSO algorithms are employed to adjust supervised training of adaptive ANN in short-term hourly load forecasting in [9]. A new class of moving filtering techniques and adaptive prediction models that are specifically designed to deal with runtime and short-term forecast of time series, which originate from monitors of system resources of Internet based servers, is developed in [10]. A generalized regression neural network based on principal components analysis is used for electricity price forecasting in [11]. A technique based on Self-Organizing Map neural network and Support Vector Machine models is proposed for predicting day-ahead electricity prices in [12]. Although the search for a best time series forecasting method continues, it is agreeable that no single method will outperform all others in all situations.

It is well known that the Hankel matrix, named after Hermann Hankel, is widely used for system identification when given a sequence of output data a realization of an underlying state-space model is desired. A first solution to this challenging theoretical problem that became known as the state-space realization problem was provided in 1965 in [13]. The key tool for solving this problem is the Hankel matrix, whose factorization into the product of an observability matrix and controllability matrix is known as the Ho–Kalman realization method [13]. The Hankel matrix-based models are appropriate to describe linear input/output mappings by infinitely many parameters, in general, since they might be obtained directly from available input/output data on the system.

* Corresponding author. Tel.: +370 69822456; fax: +370 37330446.
E-mail addresses: minvydas.ragulskis@ktu.lt (M. Ragulskis), kristina.lukoseviciute@ktu.lt (K. Lukoseviciute), zenonas.navickas@ktu.lt (Z. Navickas), rita.palivonaite@ktu.lt (R. Palivonaite).
URL: http://www.personalas.ktu.lt/~mragul (M. Ragulskis).
[1] Tel.: +370 61847349.
[2] Tel.: +370 68223789.
[3] Tel.: +370 67287255.

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.02.017

It took years of research to go from the theoretical results described in [13] to a numerically reliable realization algorithm [14]. The combination of deterministic realization theory based on the factorization of the Hankel matrix, with the theory of Markovian and innovations representations, gave rise to the stochastic theory of minimal realizations. The stochastic realization problem was studied intensively during the early 1970s in connection with innovations theory and spectral factorization theory [15,16].

Many new innovative applications based on the Hankel matrix have been developed in diverse areas of science and engineering. Gathering outputs from an impulse-response simulation into a generalized Hankel matrix and its singular value decomposition (SVD) helps to obtain reduced order models for high dimensional linear dynamical systems [17]. Hankel matrix is used to expand the original time series into the trajectory matrix of the system in [18]; singular value decomposition of the trajectory matrix helps to forecast paroxysmal events. Hankel transform of an integer sequence is defined in [19] and used to classify certain integer sequences. A Hankel matrix approach is used for dynamical systems identification from time series data in [20]. All these techniques are based on the assumption that the underlying state-space realization of the system can be described by some sort of analytical model of a dynamical system.

A new approach to the identification of a numerical sequence is proposed in [21]. The concept of the Hankel rank of a sequence is proposed in [21]. It is important to note that the Hankel rank of a sequence is a concept independent from the state-space realization of the system. The Hankel rank just describes algebraic relationships between elements of the sequence without pretending to approximate the analytical model of an underlying dynamical system; moreover, these algebraic relationships are exact.

The Hankel rank is used to express solutions of nonlinear differential equations in forms comprising ratios of finite sums of standard functions [22–24]. We will exploit the Hankel rank for the identification of the skeleton algebraic progression in the time series and use this information to forecast future values of that time series. This paper is organized as follows. The concept of the algebraic progression is presented in Section 2; the forecasting strategy is developed in Section 3; computational experiments and the selection of parameters of the forecasting method is done in Section 4; computational experiments are discussed in Section 5 and concluding remarks are given in the Section 6.

2. The definition of the rank of a sequence

Let S be a sequence of real or complex numbers:

$$S := (x_0,x_1,x_2,\ldots) := (x_k; k \in Z_0) \tag{1}$$

A subsequence of S is denoted by S_j; $j=0,1,2,\ldots$:

$$(x_j,x_{j+1},x_{j+2},\ldots) := S_j \tag{2}$$

It can be noted that $S=S_0$. The Hankel matrix H can be constructed from the sequence S:

$$H := \begin{bmatrix} x_0 & x_1 & x_2 & \cdots \\ x_1 & x_2 & x_3 & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} \tag{3}$$

Minors $H_j^{(m)}$ of H are defined as follows:

$$H_j^{(m)} := [x_{r+s-2+j}]_{1 \le r,s \le m} = \begin{bmatrix} x_j & x_{j+1} & \cdots & x_{j+m-1} \\ x_{j+1} & x_{j+2} & \cdots & x_{j+m} \\ & & \cdots & \\ x_{j+m-1} & x_{j+m} & \cdots & x_{j+2m-2} \end{bmatrix} \tag{4}$$

Determinants of these minors are denoted by $d_j^{(m)}$: $\det H_j^{(m)} = d_j^{(m)}$.

Definition 1. The rank of a subsequence S_j is such that the natural number m_j satisfies the following condition (if the rank exists):

$$d_j^{(m_j+k)} = 0 \tag{5}$$

for all $k \in N$; when $d_j^{(m_j)} \neq 0$.

We will use the following notation:

$$m_j = Hr(x_j,x_{j+1},\ldots) = HrS_j \tag{6}$$

If such number m_j does not exist, we will note that the subsequence S_j does not have a rank: $HrS_j := +\infty$.

Definition 2. The rank of a sequence S is a number m_0 if only $m_0 < +\infty$:

$$HrS = m_0 \tag{7}$$

Otherwise, the sequence S does not have a rank:

$$HrS = +\infty \tag{8}$$

Comment 1. By definition we will assume that

$$Hr(0,0,0,\ldots) := 0 \tag{9}$$

Example 1. Let $S:=(1,1,1,0,0,\ldots)$. Then, $HrS=HrS_0=3$; $HrS_1=2$; $HrS_2=1$; $HrS_j=0$ for $j=3,4,\ldots$

Example 2. Let $S:=(j;\ j\in Z_0)$. Then, $d_j^{(1)} = |j| = j$; $d_j^{(2)} = \begin{vmatrix} j & j+1 \\ j+1 & j+2 \end{vmatrix} = -1$; but $d_j^{(m)} = 0$ for $m=3,4,\ldots$ for all $j \in Z_0$. Therefore $HrS=HrS_j=2$ for all $j \in Z_0$.

Example 3. Let $S:=(j!;\ j\in Z_0)$. Then, $HrS=HrS_j=+\infty$ for all $j \in Z_0$. Thus, the given sequence of factorials does not have a rank.

Theorem 1. *Let us assume that the rank of the subsequence S_j is $HrS_j=m$; $m<+\infty$. Then it is possible to construct the characteristic determinant of the subsequence S_j [21]:*

$$\Delta^{(m)}S_j(\rho) := \begin{vmatrix} x_j & x_{j+1} & \cdots & x_{j+m} \\ x_{j+1} & x_{j+2} & \cdots & x_{j+m+1} \\ & & \cdots & \\ x_{j+m-1} & x_{j+m} & \cdots & x_{j+2m-1} \\ 1 & \rho & \cdots & \rho^m \end{vmatrix} \tag{10}$$

and the characteristic algebraic equation of the subsequence S_j [21]:

$$\Delta^{(m)}S_j(\rho) = 0 \tag{11}$$

The roots of Eq. (11) can be calculated: $\rho_k \in C$; $k=1,2,\ldots,r$, where recurrence indexes of these roots are n_k; $n_k \in N$ do satisfy the equality $n_1+n_2+\cdots+n_r=m$. Then the following equality holds true:

$$x_n = \sum_{k=1}^{r} \sum_{l=0}^{n_k-1} \mu_{kl} \binom{n}{l} \rho_k^{n-l};\ \ n=j,j+1,j+2,\ldots \tag{12}$$

where coefficients $\mu_{kl} \in C$; $k=1,2,\ldots,r$ and $l=0,1,\ldots,n_k-1$ can be determined from a system of linear algebraic equations that can be formed from equalities Eq. (12), assuming the expressions of elements $x_{n1},x_{n2},\ldots,x_{nm}$ of the subsequence S_j where indexes of these elements satisfy inequalities $j \le n_1 < n_2 < \cdots < n_m < +\infty$. Moreover, such system of linear algebraic equations has one and only solution.

The rigorous proof of Theorem 1 is given in [21].

Definition 3. The set of elements $x_j, x_{j+1}, x_{j+2},\ldots,x_{j+m}$ which does satisfy Eq. (12) is called a fragment of algebraic progression.

Algebraic progressions generalize arithmetic progressions $(a_0+jd;\ j\in Z_0)$ with $Hr(a_0+jd;\ j\in Z_0)=2$ and geometric progressions $(a_0\lambda^j;\ j\in Z_0)$ with $Hr(a_0\lambda^j;\ j\in Z_0)=1$.

Definition 4. A subsequence S_j is an algebraic progression if it's all elements satisfy equalities in Eq. (12).

Corollary 1. *A random sequence does not have a rank.*

The proof is straightforward. Let us assume that a random sequence has a rank. Then, according to Theorem 1, it is an algebraic progression. Thus, the dynamics of the sequence is deterministic, which contradicts the definition of a random sequence.

Corollary 2. *Let the rank of a sequence $(x_k;\ k\in Z_0)$ is m and a sequence $(\varepsilon_k;\ k\in Z_0)$ is a random sequence. Then, $Hr(x_k+\varepsilon_k;\ k\in Z_0)=+\infty$.*

The proof is straightforward. Let us introduce a sequence $y_k:=x_k+\varepsilon_k;\ k\in Z_0$. Let us assume that the rank of this sequence exists: $Hr(y_k;\ k\in Z_0)=m_1<+\infty$. Now, let us construct the sequence $((y_k-x_k);\ k\in Z_0)$. The multiplication of an algebraic sequence by a finite scalar or the summation of two algebraic sequences produces an algebraic sequence [21]. Thus, the sequence $((y_k-x_k);\ k\in Z_0)$ is an algebraic sequence, which contradicts the definition of a random sequence.

3. The forecasting strategy

Let us assume that $2n+1$ observations are available for building a model of the process and then using this model to extrapolate the past behavior into the future:

$$x_0,x_1,x_2,\ldots,x_{2n-1},x_{2n} \tag{13}$$

where x_{2n} is the value of the observation at the present moment. Having an uneven number of observations one can construct a Hankel minor $H_0^{(n+1)}$:

$$H_0^{(n+1)}=\begin{bmatrix} x_0 & x_1 & \cdots & x_n \\ x_1 & x_2 & \cdots & x_{n+1} \\ & & \cdots & \\ x_n & x_{n+1} & \cdots & x_{2n} \end{bmatrix} \tag{14}$$

Let us assume that $\det H_0^{(n+1)}\neq 0$ (if $\det H_0^{(n+1)}=0$ then the sequence could be an algebraic progression and the identification of x_{2n+1} would be straightforward). How one could build a model of the process using Eq. (12) if the sequence in Eq. (13) is not an algebraic progression?

Let us make another assumption. Let the sequence in Eq. (13) is produced by adding noise to an algebraic progression. In other words, we make a proposition that

$$x_k:=\tilde{x}_k+\varepsilon_k;\quad k=0,1,2,\ldots,2n \tag{15}$$

where $\varepsilon_k;\ k=0,1,2,\ldots,2n$ is an additive noise, and

$$\det\tilde{H}_0^{(n+1)}=\det\begin{bmatrix} x_0-\varepsilon_0 & x_1-\varepsilon_1 & \cdots & x_n-\varepsilon_n \\ x_1-\varepsilon_1 & x_2-\varepsilon_2 & \cdots & x_{n+1}-\varepsilon_{n+1} \\ & & \cdots & \\ x_n-\varepsilon_n & x_{n+1}-\varepsilon_{n+1} & \cdots & x_{2n}-\varepsilon_{2n} \end{bmatrix}=0 \tag{16}$$

here

$$\tilde{H}_j^{(m)}:=[x_{r+s-2+j}-\varepsilon_{r+s-2+j}]_{1\le r,s\le m} \tag{17}$$

Moreover, we assume that the infinite sequence $\tilde{x}_k;\ k=0,1,2,\ldots$ is an algebraic progression and this sequence is some sort of a skeleton sequence determining the global dynamics of the time series.

Natural is the question—how this additive noise $\varepsilon_k;\ k=0,1,2,\ldots,2n$ can be identified?. Elementary reasoning indicates that there exists an infinite number of solutions to this problem. Of course, the goal is to minimize any distortions from the original time series. Therefore, we introduce the fitness function for the set of corrections $\{\varepsilon_0,\varepsilon_1,\ldots,\varepsilon_{2n}\}$, which has to be maximized:

$$F(\varepsilon_0,\varepsilon_1,\ldots,\varepsilon_{2n})=\frac{1}{a\left|\det(\tilde{H}_0^{(n+1)})\right|+\sum_{k=0}^{2n}\lambda_k|\varepsilon_k|};\quad a>0 \tag{18}$$

where

$$\lambda_k=\frac{\exp(b(k+1))}{\sum_{j=0}^{2n}\exp(b(j+1))};\ \ k=0,1,\ldots,2n;\ \ b>0 \tag{19}$$

It is clear that $\sum_{k=0}^{2n}\lambda_k=1;\ 0<\lambda_0<\lambda_1<\cdots<\lambda_{2n-1}<\lambda_{2n}$.

If $\det\tilde{H}_0^{(n+1)}=0$, the fitness function reaches its maximum at $\varepsilon_0=\varepsilon_1=\cdots=\varepsilon_{2n}=0$ and $F(0,0,\ldots,0)=+\infty$ then. The parameter a determines the penalty proportion between the magnitude of the determinant and the sum of weighted corrections (both penalties have the same weight when $a=1$). Coefficients $\lambda_0,\lambda_1,\ldots,\lambda_{2n}$ determine the tolerance corridor for corrections $\varepsilon_0,\varepsilon_1,\ldots,\varepsilon_{2n}$. All corrections would have the same weight if $b=0$. The larger the b is, the higher is the weight for the correction of the observation at the present moment compared to past moments. In other words, the toleration of changes for the present moment is smaller compared to the toleration of changes for past moments. That corresponds to the supposition that the importance of the observation depends on its closeness to the present moment.

It is clear that the identification of the set of corrections $\{\varepsilon_0,\varepsilon_1,\ldots,\varepsilon_{2n}\}$ is not a straightforward computational task. First of all it should be noted that the above stated problem does not have a unique solution. Of course, corrections must be minimal, but we also allow the determinant of the corrected Hankel matrix to be not exactly equal to zero (Eq. (18)). In other words, we compute pseudoranks of the corrected sequence and try to balance between the magnitude of the determinant and the corrections. Soft computing techniques are exploited for the determination of the optimal set of corrections based on the fitness function described by Eq. (18).

As soon as the sequence is corrected, one can use Eq. (12) to compute future values of the sequence (even if the determinant of the corrected sequence is not strictly equal to zero).

It can be noted that the presented forecasting technique (though based on the Hankel matrices) does not use the concept of a state-space model, or the concept of a dynamical system. This is a nonlinear algebraic technique because the concept of Hankel rank comprises nonlinear algebraic relationships. In other words, our method uses a nonlinear algebraic identification technique of skeleton sequences. Another important feature of the presented forecasting technique is the necessity to identify the base fragment of the sequence. In other words, the parameter n must be fixed before the correction procedure is commenced. It is clear that corrections can be executed for sequences of different lengths. The goal is to select such sequence length (the base fragment) that the required corrections would be minimal. These questions, together with detailed discussions on the soft computing strategies, are presented in the next sections.

4. Computational experiments with a test time series.

We will exploit evolutionary algorithms (EA) and particle swarm optimization (PSO) algorithms for the identification of the near-optimal set of corrections $\{\varepsilon_0,\varepsilon_1,\ldots,\varepsilon_{2n}\}$. But initially we construct an artificial test time series that will be used to tune

parameters of these algorithms. First a periodic sequence is formed (numerical values of seven elements in a period are selected as 0.5; 0.7; 0.1; 0.9; 0.3; 0.2 and 0.8); this sequence represents a skeleton algebraic sequence. Next, we add random numbers uniformly distributed in the interval $[-0.15;0.15]$ to all elements of that sequence (Fig. 1). This test time series will be used for testing the functionality of the proposed forecasting method.

The first task is to identify the base fragment of the time series, which will be used to reconstruct the skeleton algebraic sequence. This would be a trivial problem for the constructed periodic sequence (the H-rank would be equal to 8), but the H-rank must be identified for the time series with the additive noise. Fig. 2 represents the relationship between the absolute value of the determinant of the Hankel matrix and the dimension of the Hankel matrix. It can be seen that the determinant is not equal to zero when the dimension is 8 (due to the additive noise). Nevertheless, we fix the dimension of the Hankel matrix to 8 ($n=7$; the length of the base fragment of the time series is $2n+1=15$). The functionality of the time series forecasting method for other dimensions of the Hankel matrix (for the same time series) will be discussed in the following sections.

4.1. The selection of parameters for the PSO algorithm

PSO is an evolutionary computation technique based on the simulation of social behavior, first introduced by Eberhart and Kennedy in 1995 [25]. Through cooperation and competition among the population, PSO optimization approaches can find good solutions efficiently and effectively. Each individual in PSO is treated as a volume-less particle (a material point) in the D-dimensional space (we will use $D=2n+1$). Each particle in PSO flies in the search space with a velocity that is dynamically adjusted both to its own flying experience and the group's flying experience. The ith particle is represented by its coordinates as $X_i=(x_{i1},x_{i2},\ldots,x_{iD})$, $i=1,2,\ldots,$ m where m is the population's size. The previous position giving the best fitness value of the ith particle in its flight trajectory is recorded and represented as $P_i=(p_{i1},p_{i2},\ldots,p_{iD})$. The index of the best particle among all particles in the population is represented by symbol g. The velocity of the ith particle is represented as $V_i=(v_{i1},v_{i2},\ldots,v_{iD})$. Eq. (20) describes the velocity and position update equations for the PSO population:

$$v_{id}=w\,v_{id}+c_1\,r_1(p_{id}-x_{id})+c_2\,r_2(p_{gd}-x_{id});$$
$$x_{id}=x_{id}+v_{id};\ i=1,2,\ldots,D \tag{20}$$

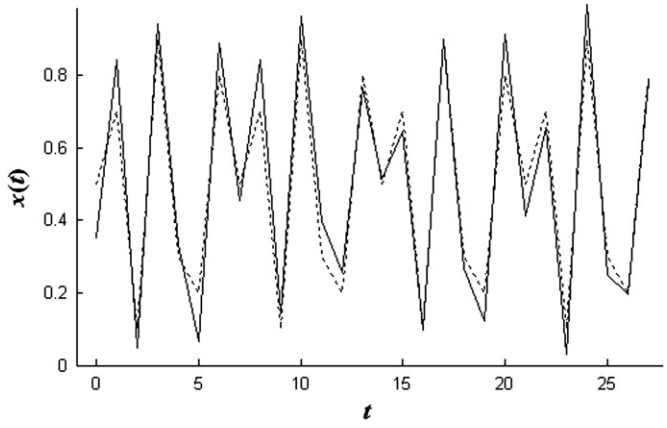

Fig. 1. The test time series (the solid line) and the periodic time series (the dashed line).

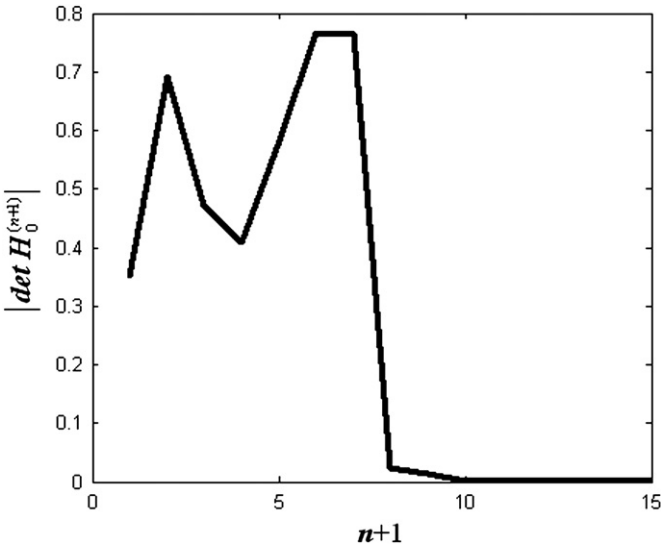

Fig. 2. The relationship between the absolute value of the determinant of the Hankel matrix and its order.

where r_1 and r_2 are two random variables distributed evenly in the interval [0,1]; c_1 and c_2 are two positive constants, called acceleration constants, representing weightings of the stochastic acceleration terms that pull each particle toward the *particle's best* and the *global best*; and w is the inertia weight balancing the global and the local search.

Early experience with PSO optimization (trial and error, mostly) led to set acceleration constants c_1 and c_2 equal to 2.0 and $w=1$ for almost all applications [26]. The inertia weight was brought in to control the balance between the global and the local exploration abilities in [27]; the recommendation was to use $w=0.9$. A large inertia parameter facilitates a global search, while a small inertia parameter facilitates a local search.

In 1999 Clerc indicated that the use of the constriction factor K (Eq. (21)) may be necessary to ensure convergence of the PSO [28]:

$$v_{id}=K(v_{id}+c_1\,r_1(p_{id}-x_{id})+c_2\,r_2(p_{gd}-x_{id})) \tag{21}$$

where $K=(2/(|2-\varphi-\sqrt{\varphi^2-4\varphi}|))$ and $\varphi=c_1+c_2$; $\varphi>4$.

In all cases, where Clerc's constriction method was used, φ was set to 4.1 and the constant multiplier K is thus 0.729. Shi and Eberhart [29] noted that this is equivalent to using Eq. (20) with $w=0.729$ and $c_1=c_2=1.494$.

Convergence analysis and stability studies of PSO have been reported by Trelea [30], where the inertia weight 0.6 with $c_1=c_2=1.7$ outperformed the selection of parameters recommended by Shi and Eberhart.

Anyway, despite numerous research efforts, the selection of the parameters remains mostly empirical and depends on the topology of the target function and/or on the structure of the fitness function. We have selected three sets of the PSO parameters according to the recommendations in [26,28,30]: the parameter set 1 with $w=0.9$ and $c_1=c_2=2$; the parameter set 2 with $w=0.729$ and $c_1=c_2=1.494$ as recommended by Clerc; and the parameter set 3 with $w=0.6$ and $c_1=c_2=1.7$ as recommended by Trelea.

There have been no definitive recommendations in the literature regarding the swarm size in PSO. Eberhart and Shi indicated that the effect of the population size on the performance of the PSO method is of minimum significance. Most researchers use a swarm size of 10–60, but there are no established guidelines [31]. For the purpose of comparing PSO and GA efficiency, the swarm size that is used for PSO is the same as the population size in their

equivalent GA. PSO swarm size is fixed to 50 particles compared to 50 chromosomes in the GA population.

4.2. Computational experiments with PSO

As mentioned previously, the goal of optimization algorithms is to maximize the fitness function defined by Eq. (18). We fix $a=1$ in Eq. (18), thus equalizing the balance between the magnitude of the determinant of the Hankel matrix and the magnitude of corrections.

In order to clarify the procedure of the computation of a single forecast, we present a detailed sequence of computations (with the parameter set 3 and $b=0.5$).

Initially, $2n+1=15$ values of λ_k are computed according to Eq. (18) (the first column in Table 2). The skeleton of the test time series is presented in the second column; the test time series itself—in the third column of Table 2. A single execution of PSO algorithm produces a set of corrections listed in the fourth column of Table 2. The characteristic equation then reads

$$\det\tilde{H}_0^{(8)} = \det\begin{bmatrix} 0.2381 & 0.9879 & 0.1422 & 0.9229 & 0.4330 & 0.1523 & 0.8345 & 0.3684 \\ 0.9879 & 0.1422 & 0.9229 & 0.4330 & 0.1523 & 0.8345 & 0.3684 & 0.8105 \\ 0.1422 & 0.9229 & 0.4330 & 0.1523 & 0.8345 & 0.3684 & 0.8105 & 0.1476 \\ 0.9229 & 0.4330 & 0.1523 & 0.8345 & 0.3684 & 0.8105 & 0.1476 & 1.0165 \\ 0.4330 & 0.1523 & 0.8345 & 0.3684 & 0.8105 & 0.1476 & 1.0165 & 0.3975 \\ 0.1523 & 0.8345 & 0.3684 & 0.8105 & 0.1476 & 1.0165 & 0.3975 & 0.2700 \\ 0.8345 & 0.3684 & 0.8105 & 0.1476 & 1.0165 & 0.3975 & 0.2700 & 0.7716 \\ 1 & \rho^1 & \rho^2 & \rho^3 & \rho^4 & \rho^5 & \rho^6 & \rho^7 \end{bmatrix} = 0 \quad (22)$$

It is clear that a new set of near-optimal corrections $\{\varepsilon_0,\varepsilon_1,\ldots,\varepsilon_{2n}\}$ is generated every time when the PSO algorithm is executed. Thus we execute PSO algorithm 100 times, compute the forecasted value of x_{15} (100 different estimates of x_{15} are produced in the process) and calculate root mean square errors (RMSE) between the true value of x_{15} and 100 forecasted estimates of x_{15} (Fig. 3). Results of computational experiments are presented in the first three rows of Table 1. Best results are achieved with the third set of parameters at $b=0.5$.

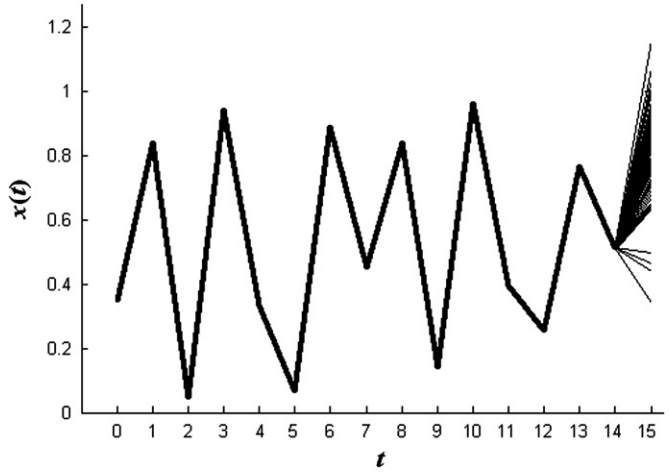

Fig. 3. The base fragment of the test time series (the thick solid line) and 100 forecasts of x_{15} (thin solid lines).

and seven roots are listed in the first column of Table 3. It can be noted that all roots are different (all recurrence indexes $n_i=1$; $i=1,2,\ldots,7$) and thus Eq. (7) takes the following form:

$$x_n = \sum_{k=1}^{7} \mu_{k0}\rho_k^n; \quad n=0,1,2,\ldots \quad (23)$$

Table 2
Weights of corrections λ_k; elements of the periodic time sequence s_k; elements of the test time series x_k; corrections ε_k; elements of the identified skeleton sequence $x_k-\varepsilon_k$; and the length of the base fragment $2n+1=15$.

k	λ_k	s_k	x_k	ε_k	$x_k-\varepsilon_k$
0	0.0003590	0.5	0.3515	0.1133	0.2381
1	0.0005919	0.7	0.8419	−0.1460	0.9879
2	0.0009759	0.1	0.0464	−0.0958	0.1422
3	0.0016089	0.9	0.9416	0.0188	0.9229
4	0.0026526	0.3	0.3353	−0.0976	0.4330
5	0.0043735	0.2	0.0663	−0.0860	0.1523
6	0.0072106	0.8	0.8884	0.0538	0.8345
7	0.0118883	0.5	0.4526	0.0841	0.3684
8	0.0196005	0.7	0.8421	0.0316	0.8105
9	0.0323158	0.1	0.1423	−0.0053	0.1476
10	0.0532798	0.9	0.9644	−0.0521	1.0165
11	0.0878435	0.3	0.3967	−0.0007	0.3975
12	0.1448294	0.2	0.2573	−0.0127	0.2700
13	0.2387833	0.8	0.7659	−0.0057	0.7716
14	0.3936871	0.5	0.5141	0.0017	0.5124

Table 1
RMSE of prediction errors produced by different methods for the artificial time series; b is the parameter determining the tolerance corridor for corrections $\varepsilon_0,\varepsilon_1,\ldots,\varepsilon_{2n}$ in the fitness function. Bold numbers identify best values of b for each of the methods.

Method	b										
	0	0.25	0.5	0.75	1	1.25	1.5	1.75	2	3	4
PSO, set 1	0.1952	**0.1950**	0.1997	0.2071	0.2030	0.2029	0.1997	0.2022	0.2010	0.2038	0.2017
PSO, set 2	0.1892	0.1912	0.1903	0.1931	0.1954	0.1999	**0.1890**	0.1957	0.1979	0.2029	0.1953
PSO, set 3	0.1889	0.1911	**0.1883**	0.1929	0.1926	0.1906	0.1953	0.1970	0.1994	0.1948	0.1955
GA, $\beta=0$	0.1899	0.1916	0.1919	0.1939	**0.1878**	0.1883	0.1886	0.1921	0.1948	0.1904	0.2061
GA, $\beta=2$	0.1870	0.1885	0.1886	0.1883	**0.1861**	0.1886	0.1893	0.1882	0.1896	0.1888	0.1893

Table 3
Roots of the characteristic equation ρ_i and coefficients μ_{i0}.

i	ρ_i	μ_{i0}
1	$-0.8856+0.3868i$	$0.2106+0.1458i$
2	$-0.8856-0.3868i$	$0.2106-0.1458i$
3	$-0.2407+1.0041i$	$0.0236-0.0738i$
4	$-0.2407-1.0041i$	$0.0236+0.0738i$
5	1.0037	0.5216
6	$0.5843+0.6880i$	$-0.0227-0.0656i$
7	$0.5843-0.6880i$	$-0.0227+0.0656i$

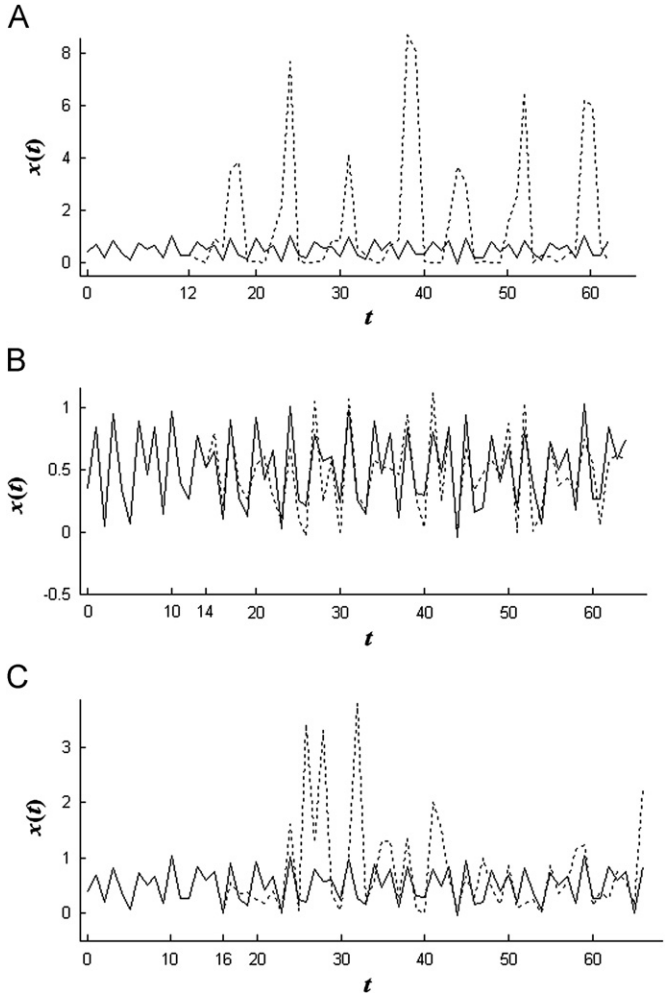

Fig. 4. Forecasts of the test time series (dashed lines) when the length of the base fragment in the algebraic sequence is underestimated ($n=6$; A), determined correctly ($n=7$; B) and overestimated ($n=8$; C).

The system of linear algebraic equations for the identification of parameters μ_{k0}; $k=1,2,\ldots,7$ reads

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \rho_1 & \rho_2 & \rho_3 & \rho_4 & \rho_5 & \rho_6 & \rho_7 \\ \rho_1^2 & \rho_2^2 & \rho_3^2 & \rho_4^2 & \rho_5^2 & \rho_6^2 & \rho_7^2 \\ \rho_1^3 & \rho_2^3 & \rho_3^3 & \rho_4^3 & \rho_5^3 & \rho_6^3 & \rho_7^3 \\ \rho_1^4 & \rho_2^4 & \rho_3^4 & \rho_4^4 & \rho_5^4 & \rho_6^4 & \rho_7^4 \\ \rho_1^5 & \rho_2^5 & \rho_3^5 & \rho_4^5 & \rho_5^5 & \rho_6^5 & \rho_7^5 \\ \rho_1^6 & \rho_2^6 & \rho_3^6 & \rho_4^6 & \rho_5^6 & \rho_6^6 & \rho_7^6 \end{bmatrix} \cdot \begin{bmatrix} \mu_{10} \\ \mu_{20} \\ \mu_{30} \\ \mu_{40} \\ \mu_{50} \\ \mu_{60} \\ \mu_{70} \end{bmatrix} = \begin{bmatrix} 0.2381 \\ 0.9879 \\ 0.1422 \\ 0.9229 \\ 0.4330 \\ 0.1523 \\ 0.8345 \end{bmatrix} \quad (24)$$

and values of parameters μ_{k0}; $k=1,2,\ldots,7$ are presented in the second column of Table 3. Finally, the forecasted value:

$$x_{15} = \sum_{k=1}^{7} \mu_{k0}\rho_k^{15} = 0.6510$$

(the actual value of the test time series is $x_{15}=0.6437$)

The forecasting horizon is fixed to 1; the window of observation is set to the base fragment of the time series. We travel with this fixed window length step by step into the future. For a fixed window of observation we forecast the next value 100 times. As soon as 100 forecasts are available (at every time step), we compute the algebraic mean of those forecasts and assume that this is the actual forecasted value. Forecasts of the test time series are shown in Fig. 4B.

So far all computational experiments have been performed with $n=7$ (the length of the base fragment of the test time series is 15). Changing the length of the base fragment is an important issue. We perform computational experiments with $n=6$ and $n=8$ (Fig. 4A and C). It can be clearly seen that our method does not work when the length of the base fragment is shorter than the minimum length of the skeleton time series required to make the determinant of its Hankel matrix equal to zero (Fig. 4A). Really, it is impossible to forecast anything if the model of the process is not adequate. On the other hand, if the length of the base fragment is longer than the minimum length of the skeleton series, the forecasting results are not so bad (Fig. 4C), but still considerably worse compared to the optimal length results. It can be noted that an algebraic sequence can be not only a periodic, arithmetic or geometric progression, but algebraic progressions also cover a much wider area of time series (another question what is the H-rank of this time series). In general, a time series is an algebraic progression if its development in time is governed by some sort of deterministic model. Thus in principle, such a concept of forecasting based on the identification of algebraic sequences should be widely applicable.

4.3. Enhancement of the forecasting by deleting most deviant trials

As mentioned previously, the forecasted value is calculated as the arithmetic mean of 100 trials. It is quite natural that some of those trials could be inaccurate due to unsuccessful allocation of the initial population, for example. One of the possibilities could be to collect all 100 trials, to interpolate the forecasted results by a Gaussian density function and then to use 3σ or 2σ rule to delete most deviant trials. Unfortunately, experiments with the test time series show that a hypothesis of Gaussian distribution cannot be statistically proved with a satisfactory reliability (that can be observed by a naked eye in Fig. 3).

Moreover, one should consider the fact that the proposed forecasting method tries to identify the skeleton algebraic sequence by correcting values of the time series in the base fragment. If those corrections are very large, the predictions can be poor simply due to the fact that the identified skeleton sequence is far away from the original time series. Therefore, we nominate most deviant trials, such trials whose forecasted numerical value is far away from the arithmetic mean of all trials, or whose sum of absolute values of corrections in the base fragment is high compared to other trials. It is clear that a specialized algorithm is necessary for the identification of most deviant trials.

We propose a rather simple algorithm that is illustrated in Fig. 5 where 100 forecasts of x_{15} are shown. The ε-axis stands for the sum of absolute values of corrections $\varepsilon_0,\varepsilon_1,\ldots,\varepsilon_{2n}$; the vertical x-axis stands for the actual forecasted value of x_{15} in each trial. Initially, the mean and the variance of the sum of corrections and the forecasts are calculated for all 100 trials. Then the scale of the

vertical axis is changed in order to equalize dispersions in mathematical coordinates in both axes (physical coordinates do not change) in Fig. 5. In other words, the shape of the cloud of trials in Fig. 5 is expanded into a more or less circular type object. Next, the mass center of the cloud is computed (it is denoted by a plus marker in Fig. 5). Then we draw such a radius from the mass center that the ratio of trials in the region limited by the circle and two horizontal lines and the total number of trials is equal to a predefined fraction (the radius r_1 corresponds to 0.5; r_2 to 0.75; r_3 to 0.9 in Fig. 5).

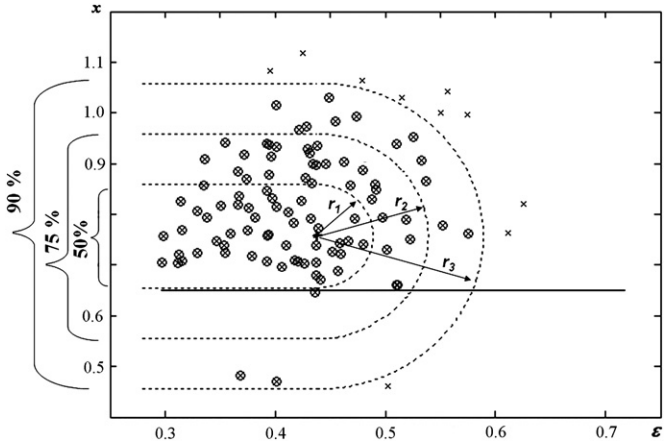

Fig. 5. A schematic diagram illustrating the deletion of the most deviant trials; ε stands for the sum of absolute values of corrections $\varepsilon_0,\varepsilon_1,\ldots,\varepsilon_{2n}$; x stands for the actual forecasted value of x_{15}; the thick cross corresponds to the mass center of the cloud of 100 trials; the thick horizontal line corresponds to the actual value of x_{15}; radii r_1, r_2 and r_3 correspond to the deletion of 50%, 25% and 10% of trials, respectively; thin crosses denote 10% deleted trials; and computational results correspond to data presented in Fig. 3.

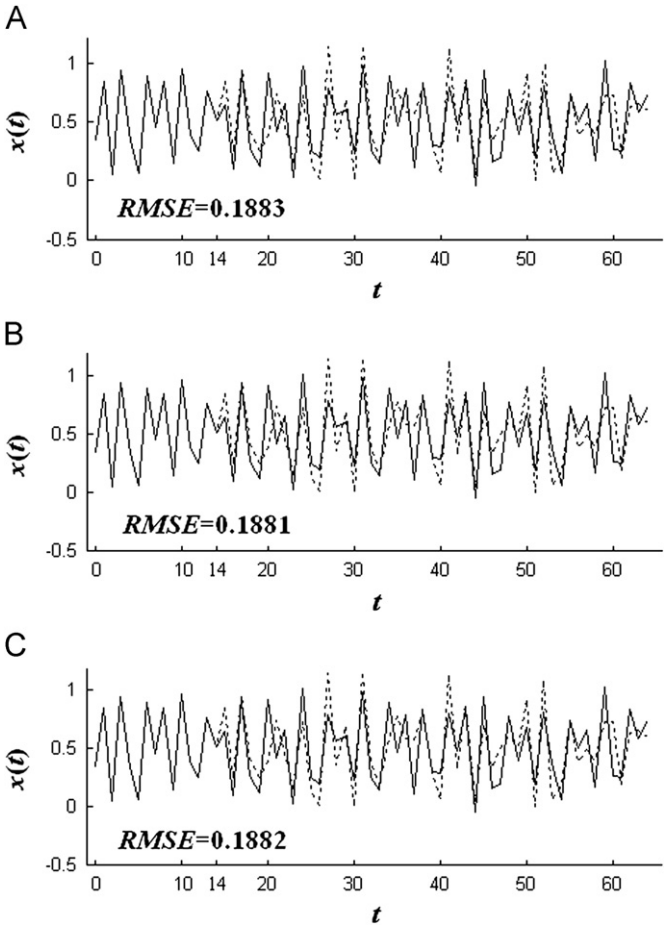

Fig. 6. Forecasts of the test time series when PSO ($n=7$; the parameter set 3; $b=0.5$) is used for the identification of corrections $\varepsilon_0,\varepsilon_1,\ldots,\varepsilon_{2n}$: A—no trials are deleted; B—10% of most deviant trials are deleted, and C—25% of most deviant trials are deleted.

Computational experiments with the test time series and PSO ($n=7$; the parameter set 3; $b=0.5$) show that the deletion of a limited number of the most deviant trials can improve the forecasting accuracy (Fig. 6).

4.4. Evolutionary algorithms versus PSO

Natural is the question if other optimization techniques could be used to identify a near-optimal set of corrections $\varepsilon_0,\varepsilon_1,\ldots,\varepsilon_{2n}$. We exploit evolutionary algorithms (EA) and compare their functionality to results produced by PSO. Though application of EA is probably not the best way to solve every problem, but some of the advantages of EA include that it deals with a large number of variables, works with experimental data and analytical functions and optimizes variables with extremely complex cost surfaces [32].

Every chromosome in our computational setup comprises a set of corrections ($2n+1$ genes represented as real numbers). The initial population comprises m chromosomes with randomly generated values of genes ($m=50$ is used in our experiments); values of genes are limited in the interval $[-0.2;0.2]$. The fitness function associated with every chromosome is defined by Eq. (18). An even number of chromosomes is selected to the mating population from the initial population. We use a random roulette method for the selection of chromosomes [33]. The higher the fitness value of a chromosome, the higher is a chance that this chromosome will be selected to the mating population. Nevertheless, a probability that a chromosome with a low fitness value will be selected is not zero. Also, several copies (clones) of the same chromosome can be selected to the mating population. All chromosomes are grouped into pairs after the selection process.

The crossover between two chromosomes in a pair is executed for all pairs in the mating population. We use a one-point modified β-crossover method [34] (the location of the point is random for every pair). We not only exchange genes, but also let two new offspring chromosomes to become more or less similar to each other and to their parent chromosomes. The β-crossover algorithm can be described by the following equations:

$$\begin{aligned}
\tau_{j,L}^{(k+1)}\Big|_{-} &:= round\left(\frac{\beta}{\beta+1}\tau_{j,L}^{(k)}+\frac{1}{\beta+1}\tau_{j,R}^{(k)}\right);\\
\tau_{j,R}^{(k+1)}\Big|_{-} &:= round\left(\frac{1}{\beta+1}\tau_{j,L}^{(k)}+\frac{\beta}{\beta+1}\tau_{j,R}^{(k)}\right);\\
\tau_{j,L}^{(k+1)}\Big|_{+} &:= round\left(\frac{1}{\beta+1}\tau_{j,L}^{(k)}+\frac{\beta}{\beta+1}\tau_{j,R}^{(k)}\right);\\
\tau_{j,R}^{(k+1)}\Big|_{+} &:= round\left(\frac{\beta}{\beta+1}\tau_{j,L}^{(k)}+\frac{1}{\beta+1}\tau_{j,R}^{(k)}\right)
\end{aligned} \tag{25}$$

where k is the generation number; $\tau_{j,L}^{(k)}$ is a jth gene of the left parent chromosome in the pair; $\tau_{j,R}^{(k)}$ is a jth gene of the right parent chromosome in the pair; $\tau_{j,L}^{(k+1)}$ is the jth gene of the left daughter chromosome; $\tau_{j,R}^{(k+1)}$ is the jth gene of the right daughter chromosome; the minus sign in the subscript denotes that the jth gene is above the crossover point; and the plus sign in the subscript denotes that the jth gene is below the crossover point. In any case, jth genes of daughter chromosomes will fit into interval $\left[\tau_{j,L}^{(k)},\tau_{j,R}^{(k)}\right]$. The β-crossover algorithm converges to the

classical crossover algorithm when β tends to zero or to infinity (both offspring chromosomes are equal when $\beta=1$).

The crossover operator is not usually applied to all pairs of chromosomes in the intermediate population [35]. A crossover coefficient κ characterizes a probability that the crossover procedure will be executed for a pair of chromosomes in the mating population. A mutation procedure is used to prevent convergence on one local solution and helps to seek the global solution. The mutation parameter μ ($0 \le \mu < 1$) determines the intensity of the mutation process [32]. We run through genes of all chromosomes in the current generation and generate a random number evenly distributed in an interval $[-0.2;0.2]$ for every gene.

4.5. The selection of parameters of EA

In general, the selection of parameters of evolutionary algorithms is an empirical process, though some common principles are described in [23,36]. The following parameters of the evolutionary algorithm must be pre-selected: the crossover coefficient κ; the mutation parameter μ; the parameter of similarity β and the number of generations. We will use recommendations for a classical model of an evolutionary algorithm [33]. The crossover coefficient κ will be selected from an interval [0.6;0.8] and the mutation parameter μ from an interval [0;0.15]. As mentioned previously, a numerical value of the parameter of domination $\beta=\infty$ (or $\beta=0$) corresponds to the classical model of evolutionary algorithms. We will investigate the interval $1 \le \beta \le 5$ instead. There are no definitive methods of establishing how many generations an evolutionary algorithm should run for. Problem may converge on good solutions after only 80 generations; we use 40 generations in our computational experiments.

A single execution of an evolutionary algorithm produces one set of corrections $\varepsilon_0,\varepsilon_1,\ldots,\varepsilon_{2n}$. Clearly, the outcome depends on the initial population of chromosomes (among other random factors). Similarly to PSO, we execute EA (at fixed values of parameters) for 100 times, calculate 100 predictions of x_{15} and compute RMSE of the prediction. We test the accuracy of the prediction by varying parameters κ, μ and β (Table 4); the best result is achieved at $\kappa=0.7$; $\mu=0.15$ and $\beta=2$. This set of parameters is fixed for further experiments.

As mentioned previously, the functionality of the PSO based prediction method can be enhanced by selecting appropriate value of the parameter b in Eq. (18), which defines the tolerance corridor for the set of corrections and deleting most deviant trials. Similar enhancements can be performed for the EA based prediction method and are described below. Computational experiments with the test time series confirm that best results are achieved at $b=1$ (Table 1) and by deleting 10% of most deviant values (Fig. 7).

In principle, the process of the identification of the algebraic sequence could be described using statistical techniques (Fig. 5). But it is important to note that we do not filter the noise out of the signal. The principle of noise removal is completely different here. The general idea is to identify the nearest algebraic sequence by making minimal deformations of the original signal (Eq. (18)). Some sort of statistical analysis is possible for the test time series (where the algebraic sequence is known beforehand). But the statistical assessment of the noise removal becomes hard when dealing with real-world time series because we do not know how the base fragment of the algebraic sequence looks like before the noise is removed. Moreover, one trial may lead to one algebraic sequence and another trial may lead to a different algebraic sequence even for the same initial conditions (Fig. 3). We map the corrections and delete the most deviant trials using previously described techniques instead. This procedure can be considered as a generalized statistical analysis though it does not require the identification of a particular probability density function describing the distribution of trials. We leave the process to evolutionary algorithms and tune parameters of these algorithms, whereas the fitness function defined by Eq. (18) (and of

Table 4
RMSE of prediction errors for the artificial time series when EA are used for the identification of corrections $\varepsilon_0,\varepsilon_1,\ldots,\varepsilon_{2n}$.

κ	μ	$\beta=0$	$\beta=1$	$\beta=2$	$\beta=3$	$\beta=4$	$\beta=5$
0.6	0.001	0.1992	0.1974	0.1973	0.1962	0.1967	0.1964
0.6	0.005	0.1960	0.1956	0.1955	0.1955	0.1946	0.1961
0.6	0.01	0.1941	0.1943	0.1939	0.1932	0.1935	0.1937
0.6	0.05	0.1910	0.1893	0.1897	0.1904	0.1912	0.1914
0.6	0.1	0.1891	0.1894	0.1888	0.1889	0.1902	0.1900
0.6	0.15	0.1903	0.1889	0.1891	0.1893	0.1903	0.1893
0.7	0.001	0.1952	0.1965	0.1970	0.1951	0.1959	0.1966
0.7	0.005	0.1963	0.1950	0.1948	0.1949	0.1951	0.1960
0.7	0.01	0.1953	0.1938	0.1937	0.1941	0.1949	0.1956
0.7	0.05	0.1927	0.1891	**0.1882**	0.1904	0.1906	0.1908
0.7	0.1	0.1904	0.1894	0.1896	0.1894	0.1911	0.1901
0.7	0.15	0.1899	0.1890	0.1888	0.1901	0.1886	0.1901
0.8	0.001	0.2000	0.1958	0.1967	0.1957	0.1965	0.1955
0.8	0.005	0.1961	0.1944	0.1957	0.1952	0.1953	0.1960
0.8	0.01	0.1935	0.1942	0.1940	0.1931	0.1948	0.1949
0.8	0.05	0.1935	0.1893	0.1910	0.1902	0.1907	0.1904
0.8	0.1	0.1889	0.1890	0.1906	0.1900	0.1896	0.1897
0.8	0.15	0.1894	0.1894	0.1893	0.1900	0.1891	0.1906

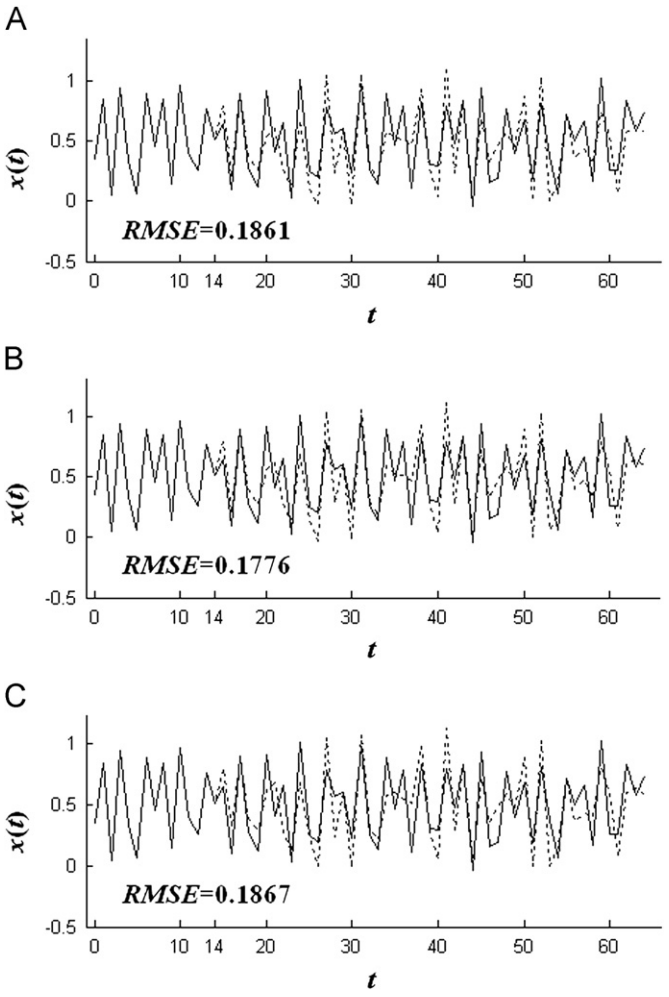

Fig. 7. Forecasts of the test time series when EA ($n=7$; $\kappa=0.7$; $\mu=0.15$; $\beta=2$; $b=1$) are used for the identification of corrections $\varepsilon_0,\varepsilon_1,\ldots,\varepsilon_{2n}$: A—no trials are deleted; B—10% of most deviant trials are deleted; C—25% of most deviant trials are deleted.

course the RMSE of the prediction) measures the performance of the method.

Finally, the overall design procedure of the proposed method can be generalized by the following structural algorithm:

(0) Identify the length of the base fragment of the skeleton algebraic sequence (the parameter n).
(1) Check if the time series is longer than the base fragment (the method cannot be used otherwise).
(2) Select the penalty proportion between the magnitude of the determinant and the sum of weighted corrections in the fitness function (we set $a=1$).
(3) Determine the tolerance corridor for corrections (we set $b=1$).
(4) Forecast the next element of the sequence x_{2n+1}.
  (4.1) Repeat 100 times:
    (4.1.1) Compute a single set of corrections $\{\varepsilon_0,\varepsilon_1,\ldots,\varepsilon_{2n}\}$ (parameters of EA are fixed to the following values: $\kappa=0.7$; $\mu=0.15$; $\beta=2$).
    (4.1.2) Construct the characteristic equation, compute its roots.
    (4.1.3) Compute a single forecast of x_{2n+1} using Eq. (12).
  (4.2) Delete most deviant trials.
  (4.3) Compute the averaged forecast of x_{2n+1}.
(5) Shift the observation window by 1 step to the right and return to step (4).

5. Computational experiments

5.1. The test time series with uniform noise

We continue computational experiments with the test time series and compare the functionality of our forecasting technique with other methods. Experiments are carried out with Box–Jenkins's time series analysis procedure autoregressive integrated moving average (ARIMA(4,1,3)) [37] as the experiments found the 4-1-3 architecture as the best model for the test time series (the order of the autoregressive part is 4; the order of the integrated part is 1 and the order of the moving average part is 3). RMSE of the ARIMA prediction is 0.1307 (Table 5; Fig. 8A) and is about 1.3 times lower compared to our method (compared to Fig. 7B). We start ARIMA predictions from the 15th element in order to make better comparisons with our method.

It is apparent that ARIMA(4,1,3) outperforms our method. Nevertheless, the ARIMA prediction (Fig. 8A) has an expressed character of a moving average technique. Our method, on the contrary, is based not on some sort of statistical algorithms but performs local individual identification of the skeleton algebraic progression for every time step. We have to admit that our predictions are quite far from the real time series in some points (what actually spoils the average assessment of the prediction quality). But local fluctuations of the predicted time series by our method give a much better representation of the character of the real time series. For instance, our method would outperform ARIMA if one would be interested to identify a day-ahead local maximum and local minimum [2].

Table 5
RMSE of forecasts of the test time series.

The prediction method	The proposed method	ARIMA(4,1,3)	MA	SES	The compound method
RMSE	0.1776	0.1307	0.2602	0.3482	0.1290

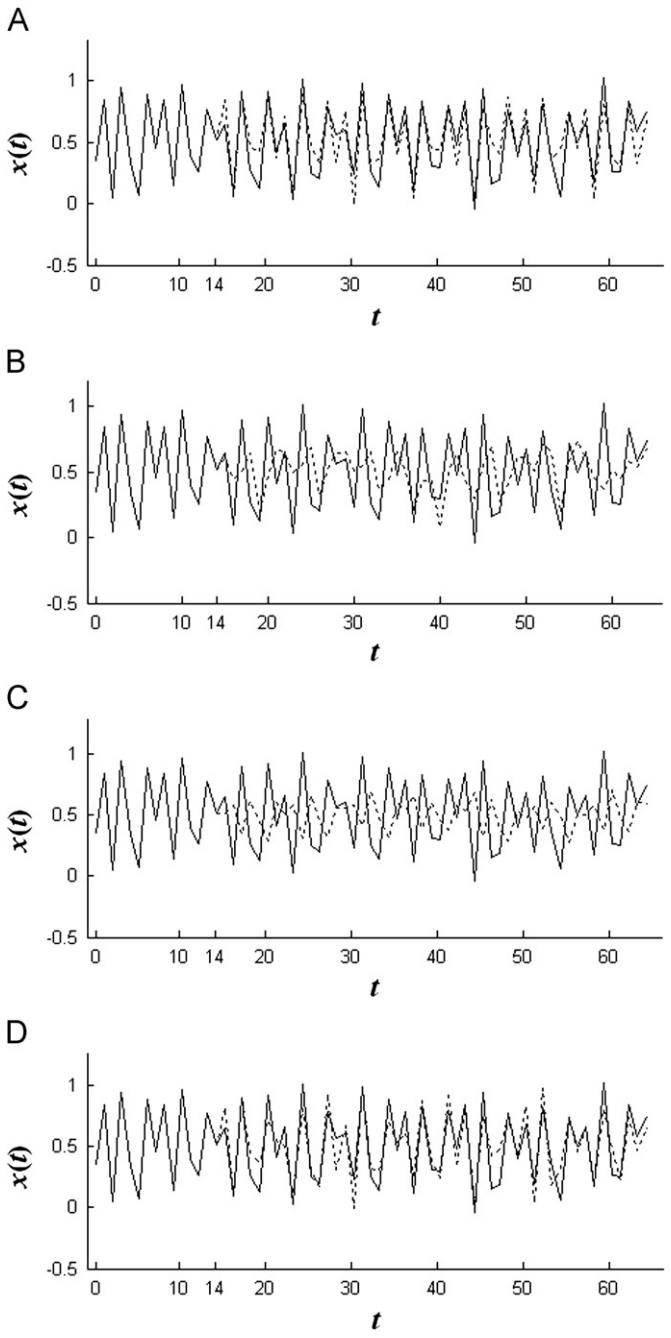

Fig. 8. Forecasts of the test time series by ARIMA(4,1,3) (A); the MA method (B); the SES method (C) and the compound method (D).

On the other hand, it can be noted that the nature of the method based on the identification of an algebraic sequence and its extrapolation to future involve much more complex analysis than a simple detection of a local maximum and a local minimum. Our method cannot be compared to a primitive technique based on bare looking at past minimum and maximum values of the time series (after the noise is minimized). Such an approach based on past minimum and maximum values could work for a periodic sequence where the period is quite short. It is clear that periodic sequences are algebraic sequences. But the set of algebraic sequences is much wider; periodic sequences are just trivial examples of algebraic sequences.

It could be possible to conciliate the moving average feature of ARIMA(4,1,3) and the local variability of our method by calculating an arithmetic mean of predictions by both methods at each

point; the results are presented in (Fig. 8D). The RMSE of such a compound prediction is better compared to ARIMA(4,1,3) (Table 5). But what is even more important, the variability of the predicted time series is much closer to the real time series.

We also compare the functionality of our method with predictors based on the moving average [38] and the exponential smoothing [39]. An often used industrial technique to remove inherent random variation in a collection of data is the simple moving average smoothing (MA) [40]:

$$S_t = \frac{1}{k}\sum_{i=0}^{k-1} x_{t-i} = S_{t-1} + \frac{x_t - x_{t-k}}{k} \tag{26}$$

where S_t is a smoothed value at the moment t; and k is a predefined constant. We set $k=2$ because experiments found that RMSE of prediction errors produced by MA method are lowest compared to other values of k. The initial prediction of x_2 is calculated from (x_0,x_1). Then we shift the observation window incrementally and compute subsequent predictions using Eq. (26). RMSE of prediction errors produced by MA is 0.2602 (Table 5); the results are presented in Fig. 8B. It is clear that simple smoothing of data cannot outperform our method in this situation.

Exponential smoothing is a simple and pragmatic approach to forecasting, whereby the forecast is constructed from an exponentially weighted average of past observations [39]. Single exponential smoothing (SES) method assigns exponentially decreasing weights as the observation gets older:

$$S_t = \alpha x_{t-1} + (1-\alpha) S_{t-1} \tag{27}$$

where $t=1,2,3,\ldots$; $S_1=x_0$ and α is the smoothing factor, $0 \le \alpha \le 1$ [38;41]. The selection of a best value of the parameter α is discussed in [39]; we choose the default value $\alpha=0.5$. A single smoothing may not excel if a trend or seasonality is present in data; double or even triple exponential smoothing may produce better results than [38]. The test series does not contain a clearly expressed trend or seasonality. Therefore we run computational experiments with a single exponential smoothing. The produced RMSE is 0.3482 (Table 5); the results are presented in Fig. 8C.

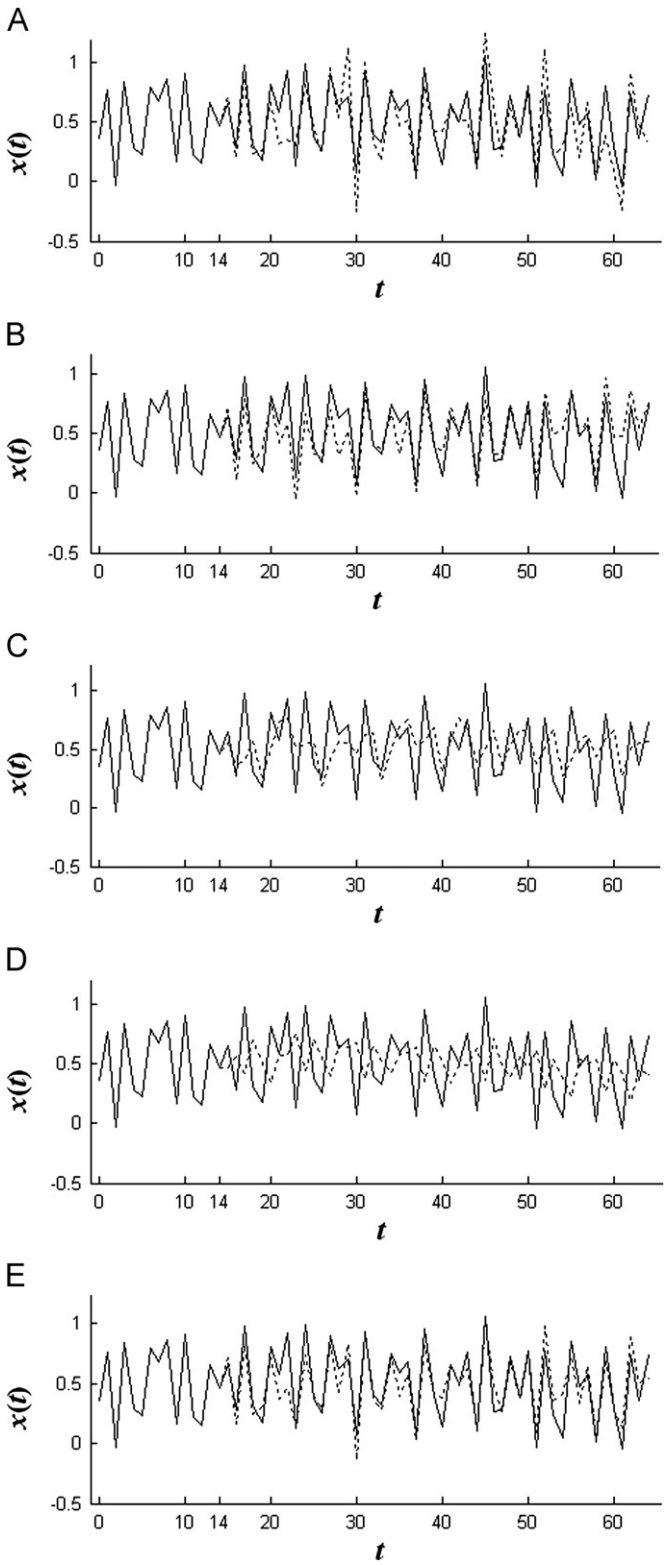

Fig. 9. Forecasts of the test time series with Gaussian noise by the proposed method (A); ARIMA(5,0,3) (B); the MA method (C); the SES method (D) and the compound method (E).

5.2. The test time series with Gaussian noise

Computational experiments are repeated with the test series, but we add Gaussian noise (zero mean and variance equal to 0.1) instead of uniform noise to the same periodic sequence. The H-rank of the updated test series is set to 8; the length of the base fragment is 15 ($n=7;2n+1=15$). Evolutionary algorithms are used to identify skeleton algebraic sequences; the same set of parameters is used as in previous experiments.

We compare the functionality of our method with ARIMA, moving average, exponential smoothing and the compound method (Fig. 9). Experiments show that the (5,0,3) architecture is the best model for ARIMA (the order of the autoregressive part is 5; the order of the integrated part is 0 and the order of the moving average part is 3) and $k=2$ is the best parameter for the MA method. RMSE for different methods are presented in Table 6. Again, ARIMA(5,0,3) outperforms our method, but the variability of our method is better. Finally it can be noted that the compound method would be a preferred one (from five methods shown in Fig. 9) if day-ahead forecasts would be considered.

5.3. Anderson dry bulb temperatures time series.

The first task is to identify the length of the base fragment of Anderson dry bulb temperatures time series [42]. The first

Table 6
RMSE of forecasts of the test time series with Gaussian noise.

The prediction method	The proposed method	ARIMA(4,0,2)	MA	SES	The compound method
RMSE	0.1771	0.1544	0.2523	0.3475	0.1288

minimum of the determinant of the Hankel matrix is reached when the dimension of the Hankel matrix is equal to 11 ($n=10$; the length of the base fragment of the time series is $2n+1=21$). We fix this length of the base fragment for the whole time series,

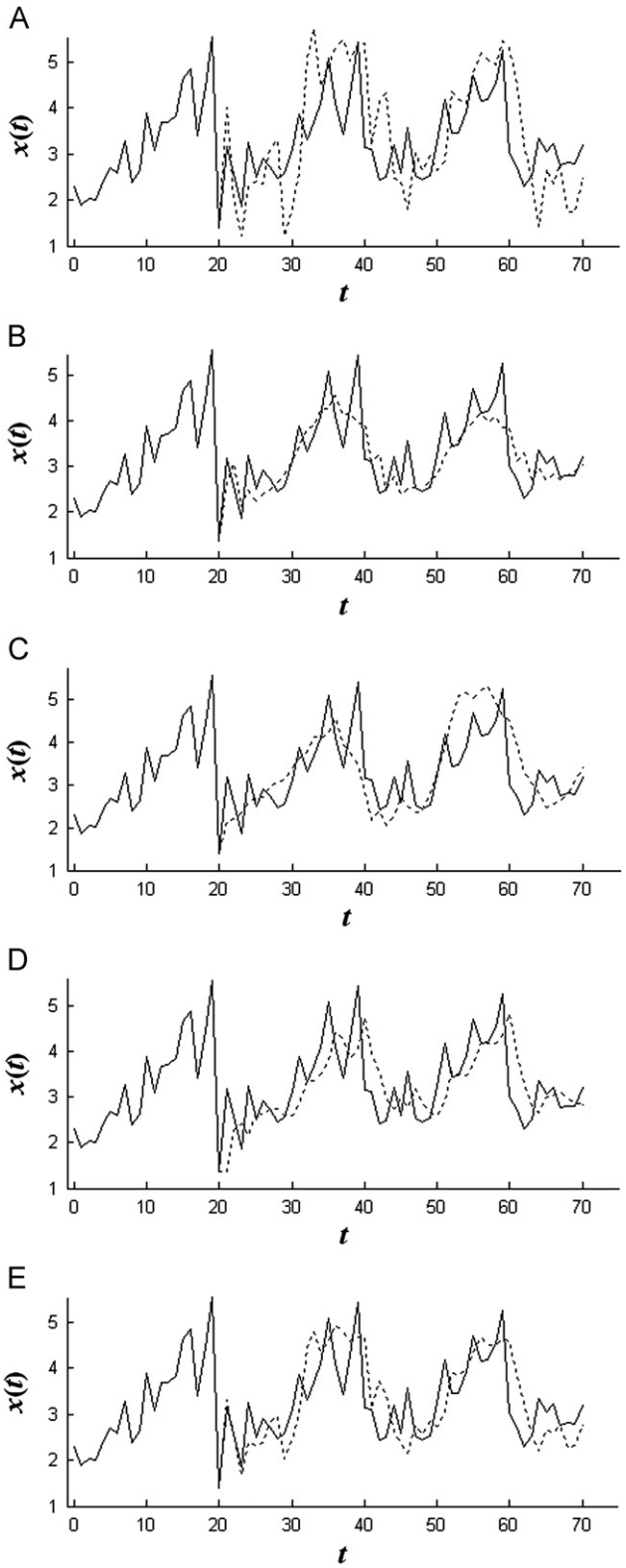

Fig. 10. Forecasts of the Anderson dry bulb temperatures time series by the proposed method (A); ARIMA(3,0,3) (B); the MA method (C); the SES method (D) and the compound method (E).

Table 7
RMSE of forecasts of the Anderson dry bulb temperatures time series.

The prediction method	The proposed method	ARIMA(3,0,3)	MA	SES	The compound method
RMSE	0.9204	0.4556	0.6321	0.6117	0.6079

keep the parameters of the EA (selected for the test time series) and perform iterative one-step forward predictions (Fig. 10A).

Experiments show that the (3,0,3) architecture is the best model for ARIMA and $k=5$ for the MA method. Forecasting results are shown in Fig. 10; RMSE for different methods are given in Table 7. ARIMA(3,0,3) outperforms our method again. But if ARIMA(3,0,3) had an expressed character of a moving average technique, it clearly averages all upper spikes of the Anderson series. The SES method produces rather good results also, but one can observe a definite phase delay between the original time series and the forecasted series (what could be definitely a negative factor if one-step-ahead forecasts would be considered). Though the RMSE of the compound prediction is worse compared to ARIMA(3,0,3) but the variability of the predicted time series is much closer to the real time series.

5.4. Monthly basic iron production in Australia (thousand tones) January 1956–August 1995

The first minimum of the determinant of the Hankel matrix of monthly basic iron production in Australia (thousand tones) January 1956–August 1995 [42] is reached when the dimension of the matrix is equal to 6 ($n=5$; the length of the base fragment of the time series is $2n+1=11$). Parameters of the EA are kept unchanged; iterative one-step forward predictions are shown in Fig. 11A (we divide all elements of the original time series by 100).

Experiments show that the (5,1,4) architecture is the best model for ARIMA and $k=9$ for the MA method. Forecasting results are shown in Fig. 11; RMSE for different methods are given in Table 8. In general, our method performs well only in some subintervals of the original time series. It seems that the algebraic relationships governing the evolution of the time series exist only in some parts of the data. There are some places where instantaneous errors produced by our method are quite high. The performance of ARIMA(5,1,4) is better in that respect (though it underestimates local maximums and minimums). The compound method (the average between the proposed method and the ARIMA(5,1,4)) shows good variability and the predicted time series follows the dynamism of the original time series. It must be noted that the RMSE of the SES method is comparable to the RMSE of the compound method (Table 8), but still worse that the RMSE of ARIMA(5,1,4). Thus it would not be advantageous to construct a compound method by averaging predictions produced by our method and the SES method.

In general, forecasting results could be further improved if few instantaneous errors produced by our method could be decreased. As mentioned previously, one of the possible reasons causing these instantaneous errors could be an abrupt change of algebraic rules defining the evolution of the original time series (and caused by multiple seasonal and cyclic components). It can be noted that a simple trend in the time series does not influence the functionality of the method. Algebraic sequences generalize arithmetic and geometric progressions (Definition 3). Therefore the computation of the rank of a sequence produced by adding an arithmetic progression and a stationary deterministic process does not cause any problems. But we had fixed the length of the base fragment

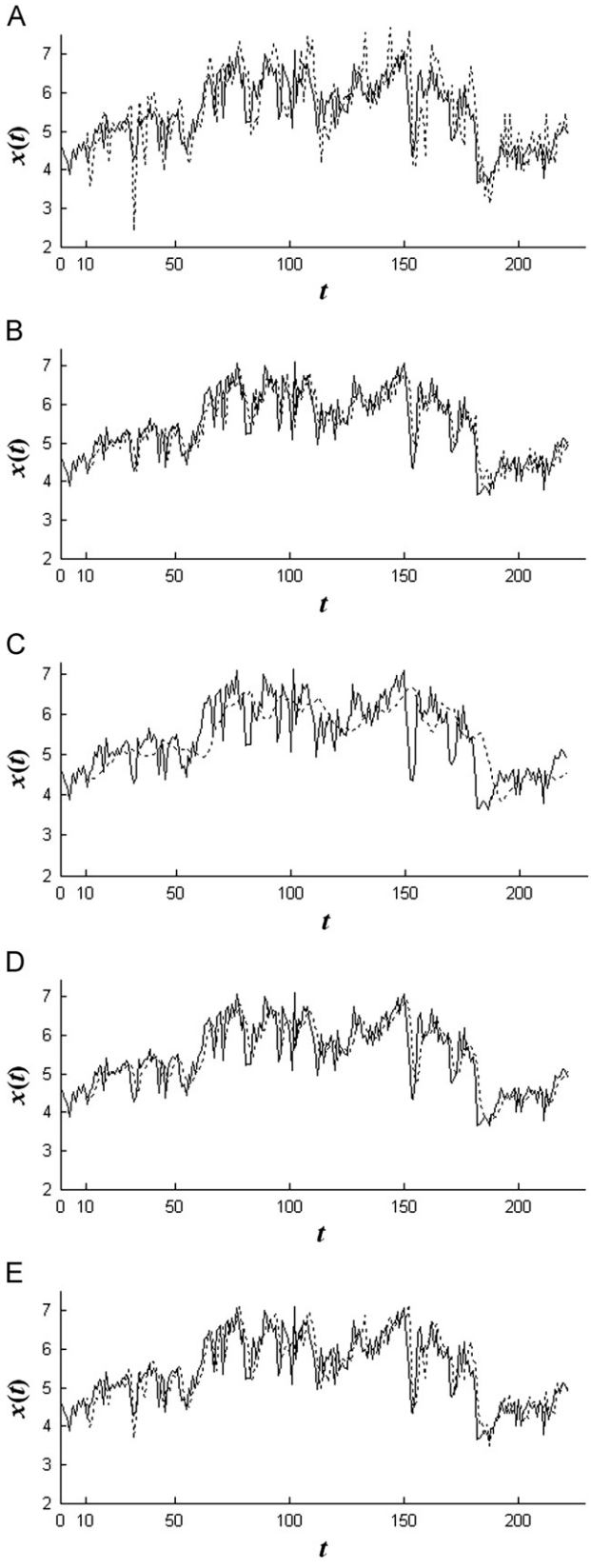

Fig. 11. Forecasts of the monthly basic iron production in Australia time series by the proposed method (A); ARIMA(5,1,4) (B); the MA method (C); the SES method (D) and the compound method (E).

for the whole time series at the beginning of the experiment. The ability of dynamical adaptation to the changing algebraic environment (adaptive variation of the length of the base fragment) could enhance the quality of the prediction by our method and is a definite objective of future research.

Table 8
RMSE of forecasts of the the monthly basic iron production in Australia time series.

The prediction method	The proposed method	ARIMA(5,1,4)	MA	SES	The compound method
RMSE	0.6727	0.4181	0.6403	0.4702	0.4759

6. Concluding remarks

A method for a short-term time series forecasting based on the identification of skeleton algebraic sequences is proposed in this paper. It can be noted that this method is especially effective when the time series is short. There are not always sufficient data to train the models, therefore such an approach when the skeleton algebraic sequence is identified from a short time series helps to extract as much information about the model of the process as possible and then use this model to extrapolate past behavior into future.

It is quite natural to expect that every method has a specific application area where its strengths are unleashed in the optimal manner. Our method is based on the proposition that a deterministic law is embedded into every time series. It is clear that our method would produce poor results for such time series where the randomness is higher that the deterministic model of the process and it becomes impossible to perform a reliable identification of algebraic sequences. Also, our method cannot be effectively used if the available time series is shorter than the minimum required length of a skeleton algebraic sequence that determines the model of the process.

It is quite probable that the forecasting accuracy of the proposed method can be improved by further advancement of algorithms used to identify the skeleton sequence. Moreover, as it is now, the length of the base fragment of the time series is fixed. Thus the presented forecasting method is based on yet another assumption that the process is stationary. Introduction of variable lengths of base fragments at different locations of the forecasted time series and advancements in the identification of the skeleton sequences remain definite targets of the future research, while the primary objective of this paper is to present the concept of an algebraic sequence and its applicability in short-term forecasting applications.

References

[1] W.R. Christiaanse, Short term load forecasting using general exponential smoothing, IEEE Transactions on Power Aparatus and Systems 90 (1971) 900–911.
[2] H.P. Satpathy, A.C. Liew, A real-time short-term peak and average load forecasting system using a self-organasing fuzzy neural network, Engineering Applications of Artificial Intelligence 11 (2) (1998) 307–316.
[3] D. Liu, D.X. Niu, M. Xing, Day-ahead price forecast with genetic-algorithm-optimized Support Vector Machines based on GARCH error calibration, Automation of Electric Power Systems 31 (11) (2007) 31–34.
[4] V. Vahidinasab, S. Jadid, A. Kazemi, Day-ahead price forecasting in restructured power systems using artificial neural networks, Electric Power Systems Research 78 (8) (2008) 1332–1342.
[5] A.L. Arciniegas, I.E.A. Rueda, Forecasting short-term power prices in the Ontario Electricity Market (OEM) with a fuzzy logic based inference system, Utilities Policy 16 (2008) 39–48.
[6] H. Kim, K. Sin, A hybrid approach based on neural networks and genetic algorithms is used for detecting temporal patterns in stock markets, Applied Soft Computing 7 (2) (2007) 569–576.

[7] J. Xue, Z. Shi, Short-time traffic flow prediction based on chaos time series theory, Journal of Transportation Systems Engineering and Information Technology 8 (5) (2008) 68–72.
[8] C.M. Lee, C.N. Ko, Time series prediction using RBF neural networks with a nonlinear time-varying evolution PSO algorithm, Neurocomputing 73 (1–3) (2009) 449–460.
[9] Z.A. Bashir, M.E. El-Hawary, Applying wavelets to short-term load forecasting using PSO-based neural networks, IEEE Transactions on Power Systems 24 (1) (2009) 20–27.
[10] S. Casolari, M. Colajanni, Short-term prediction models for server management in Internet-based contexts, Decision Support Systems 48 (1) (2009) 212–223.
[11] D.X. Niu, D. Liu, M. Xing, Electricity price forecasting using generalized regression neural network based on principal component analysis, Journal of Central South University of Technology 15 (s2) (2009) 316–320.
[12] Dongxiao Niu, Da Liu, Desheng Dash Wu, A soft computing system for a day-ahead electricity price forecasting, Applied Soft Computing 10 (3) (2010) 868–875.
[13] B.L. Ho, R.E. Kalman, Effective construction of linear state-variable models from input–output functions, Regelungstechnik 14 (12) (1966) 545–592.
[14] L.S. de Jong, Numerical aspects of recursive realization algorithms, SIAM Journal on Control and Optimization 16 (1978) 646–659.
[15] H. Akaike, Stochastic theory of minimal realization, IEEE Transactions on Automatic Control 19 (1974) 667–674.
[16] M. Gevers, T. Kailath, An innovations approach to least-squares estimation, part vi: discrete-time innovations representations and recursive estimation, IEEE Transactions on Automatic Control 18 (1973) 588–600.
[17] J. Juang, R. Pappa, An eigensystem realization algorithm for modal parameter identification and model reduction, Journal of Guidance, Control and Dynamics 8 (1985) 620–627.
[18] R. Carniel, F. Barazza, M. Tarraga, R. Ortiz, On the singular values decoupling in the Singular Spectrum Analysis of volcanic tremor at Stromboli, Natural Hazards and Earth System Sciences 6 (2006) 903–909.
[19] J.W. Layman, The Hankel transform and Some of its PRoperties, Journal of Integer Sequences 4 (01.1.5) (2001) 1–11.
[20] D. Lai, G. Chen, Dynamical systems identification from time-series data: a Hankel matrix approach, Mathematical and Computer Modelling 24 (1996) l–10.
[21] Z. Navickas, L. Bikulciene, Expressions of solutions of ordinary differential equations by standard functions, Mathematical Modeling and Analysis 11 (2006) 399–412.
[22] Z. Navickas, M. Ragulskis, How far one can go with the Exp-function method?, Applied Mathematics and Computation 211 (2009) 522–530.
[23] Z. Navickas, L. Bikulciene, M. Ragulskis, Generalization of Exp-function and other standard function methods, Applied Mathematics and Computation 216 (2010) 2380–2393.
[24] Z. Navickas, M. Ragulskis, L. Bikulciene, Be careful with the Exp-function method—additional remarks, Communications in Nonlinear Science and Numerical Simulations 15 (2010) 3874–3886.
[25] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, IEEE Service Center, Piscataway, NJ, 1995, pp. 39–43.
[26] R.C. Eberhart, J. Kennedy, Particle swarm optimization: developments, applications and resources, in: Proceedings of IEEE Congress on Evolutionary Computation, Seoul, Korea, IEEE Service Center, Piscataway, NJ, 2000, pp. 81–86.
[27] Y.H. Shi, R.C. Eberhart, Parameter selection in particle swarm optimization, in: Proceedings of Seventh Annual Conference on Evolutionary Programming, San Diego, CA, Springer-Verlag, New York, NY, 1998, pp. 591–600.
[28] M. Clerc, The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, in: Proceedings of IEEE Congress on Evolutionary Computation, Washington, DC, IEEE Service Center, Piscataway, NJ, 1999, pp. 1951–1957.
[29] Y.H. Shi, R.C. Eberhart, Comparing inertia weight and constriction factors in particle swarm optimization, in: Proceedings of IEEE Congress on Evolutionary Computation, San Diego, CA, IEEE Service Center, Piscataway, NJ, 2000, pp. 84–88.
[30] I.C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Information Processing Letters 85 (6) (2003) 317–325.
[31] J. Kennedy, R.C. Eberhart, Y.H. Shi, Swarm Intelligence, Morgan Kaufman, San Francisco, CA, 2001.
[32] R.L. Haupt, S.E. Haupt, Practical Genetic Algorithms, John Wiley & Sons, Inc., Hoboken, New Jersey, 2004.
[33] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Application to Biology, Control, and Artificial Intelligence, The MIT Press, 1992.
[34] K. Lukoseviciute, M. Ragulskis, Evolutionary algorithms for the selection of time lags for time series forecasting by fuzzy inference systems, Neurocomputing 73 (2010) 2077–2088.
[35] F. Herrere, M. Lozano, J.L. Verdegay, Tackling real-coded genetic algorithms: operators and tools for behavioural analysis, Artificial Intelligence Review 12 (4) (1998) 265–319.
[36] D. Whitley, A genetic algorithm tutorial, Statistics and Computing 4 (2) (1994) 65–85.
[37] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, Time Series Analysis: Forecasting and Control, Prentice-Hall, Englewood Clis, NJ, 1994.
[38] 〈http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc42.htm〉.
[39] J.W. Taylor, Smooth transition exponential smoothing, Journal of Forecasting 23 (2004) 385–394.
[40] 〈http://lorien.ncl.ac.uk/ming/filter/filewma.htm〉.
[41] E.S. Gardner, Jr., Exponential smoothing: the state of the art—Part II, http://www.bauer.uh.edu/gardner/docs/pdf/Exponential-Smoothing.pdf, 2005.
[42] 〈http://www.robjhyndman.com/TSDL/〉.

Ragulskis Minvydas received the Ph.D. degree in 1992 from Kaunas University of Technology, Lithuania. Since 2002 he is a professor at the Department of Mathematical Research in Systems, Kaunas University of Technology. His research interests include nonlinear dynamical systems and numerical analysis. He is the founder and head of the Research Group for Mathematical and Numerical Analysis of Dynamical Systems (http://www.personalas.ktu.lt/~mragul).

Lukoseviciute Kristina received the M.Sc. degree in mathematics in 2006 from the University of Technology, Lithuania. She is currently an assistant lecturer and a Ph.D. degree student within the Research Group for Mathematical and Numerical Analysis of Dynamical Systems, Kaunas University of Technology. Her current areas of research interests are neuro-fuzzy systems and time series forecasting.

Zenonas Navickas received the Ph.D. degree in mathematics in 1973 from Vilnius University, Lithuania. He is currently a professor at the Department of Applied Mathematics, Kaunas University of Technology. His research interests include operator analysis and differential equations.

Rita Palivonaite received the M.Sc. degree in mathematics in 2008 from the University of Technology, Lithuania. She is currently an assistant lecturer and a Ph.D. degree student within the Research Group for Mathematical and Numerical Analysis of Dynamical Systems, Kaunas University of Technology. Her current areas of research interests are combinatorial optimization and scientific visualization.