

# Short-term time series algebraic forecasting with internal smoothing



Rita Palivonaite<sup>a,1</sup>, Minvydas Ragulskis<sup>b,\*</sup>

<sup>a</sup> Research Group for Mathematical and Numerical Analysis of Dynamical Systems, Kaunas University of Technology, Studentu 50-325, Kaunas LT-51368, Lithuania

<sup>b</sup> Research Group for Mathematical and Numerical Analysis of Dynamical Systems, Kaunas University of Technology, Studentu 50-222, Kaunas LT-51368, Lithuania

## ARTICLE INFO

### Article history:

Received 30 April 2012

Received in revised form

15 April 2013

Accepted 17 August 2013

Communicated by Bijaya Ketan Panigrahi

Available online 17 October 2013

### Keywords:

Hankel matrix

Time series forecasting

Algebraic sequence

Evolutionary algorithms

## ABSTRACT

A new algebraic forecasting method with internal smoothing is proposed for short-term time series prediction. The concept of the  $H$ -rank of a sequence is exploited for the detection of a base algebraic fragment of the time series. Evolutionary algorithms are exploited for the identification of the set of corrections which are used to perturb the original time series. The proposed forecasting method is constructed to find a near-optimal balance between the variability of algebraic predictors and the smoothness of averaging methods. Numerical experiments with an artificially generated and real-world time series are used to illustrate the potential of the proposed method.

© 2013 Elsevier B.V. All rights reserved.

## 1. Introduction

Time series prediction is a challenging problem in many fields of science and engineering. Conditionally, time series prediction could be classified into long-term time series prediction techniques and short-term time series prediction techniques. In general, the object of long-term time series prediction techniques is to build a model of the process and then use this model to extrapolate past behavior into relatively long future horizons. The main objective of short-term time series prediction techniques remains the same—to build the model of the process and to project this model into the future. Unfortunately, the applicability of techniques for building the model of a long time series for a short time series is often impossible simply due to the fact that the amount of available data is too small. On the other hand, one step-forward future horizon (sometimes accompanied by the estimates of local minimums and local maximums) is a sufficient requirement for a short-term time series predictor.

Short-term time series 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 are 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 of 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 are proposed for predicting day-ahead electricity prices in [12].

An algebraic prediction technique based on the Hankel rank for the identification of the skeleton algebraic sequences in short-term time series is developed in [13]. Such an approach is used to extract information about the algebraic model of the process and then to use this model to extrapolate past behavior into future. It has been demonstrated in [13] that such algebraic predictor can be effectively used for the estimation of local minimums and

\* Corresponding author. Tel.: +370 69822456; fax: +370 37330446.

E-mail addresses: [rita.palivonaite@ktu.lt](mailto:rita.palivonaite@ktu.lt) (R. Palivonaite), [minvydas.ragulskis@ktu.lt](mailto:minvydas.ragulskis@ktu.lt) (M. Ragulskis).

URL: <http://www.personalas.ktu.lt/~mragul> (M. Ragulskis).

<sup>1</sup> Tel.: +37067287255.

maximums in day-ahead forecasting applications. It is agreeable that no single method will outperform all others in all situations. It is shown in [13] that the proposed predictor is outperformed (for some real-world time series) by such simple standard techniques as the moving average or the exponential smoothing method—if only the averaged prediction errors are considered.

The main objective of this paper is to enhance the algebraic predictor proposed in [13] by modifying the procedure for the identification of the skeleton algebraic sequences. Our goal is to employ the procedure of internal smoothing which should enable reaching a healthy balance between excellent variability of skeleton algebraic sequences and valuable properties of predictors based on the moving averaging method. The goal is to develop such a predictor which could produce reliable forecasts for very short time series—in situations when the available data is not enough for such predictors as ARIMA or short term time series nonlinear forecasting methods such as neural networks or support vector machines.

This paper is organized as follows. The concept of the Hankel rank is reviewed in Section 2; the forecasting strategy is developed in Section 3; parameter selection for evolutionary algorithms is done in Section 4; computational experiments are discussed in Section 5 and concluding remarks are given in the last section.

## 2. Preliminaries

Let a sequence of real numbers is given:

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

The Hankel matrix (the catelecticant matrix with constant skew diagonals)  $H^{(n)}$  constructed from the elements of this sequence is defined as follows:

$$H^{(n)} := \begin{bmatrix} x_0 & x_1 & \dots & x_{n-1} \\ x_1 & x_2 & \dots & x_n \\ \dots & \dots & \dots & \dots \\ x_{n-1} & x_n & \dots & x_{2n-2} \end{bmatrix} \tag{2}$$

where the index  $n$  denotes the order of the square matrix. The determinant of the Hankel matrix is denoted by  $d^{(n)} = \det H^{(n)}$ ;  $n \geq 1$ . The rank of the sequence  $(x_k; k \in Z_0)$  is such natural number  $m = Hr(x_k; k \in Z_0)$  that satisfies the following condition [13]:

$$d^{(m+k)} = 0 \tag{3}$$

for all  $k \in N$ ; but  $d^{(m)} \neq 0$ . Let us assume that the rank of the sequence is  $Hr(x_k; k \in Z_0) = m$ ;  $m < +\infty$ . Then the following equality holds true [13]:

$$x_n = \sum_{k=1}^r \sum_{l=0}^{n_k-1} \mu_{kl} \binom{n}{l} \rho_k^{n-l}; n = j, j+1, j+2, \dots \tag{4}$$

where characteristic roots  $\rho_k \in C$ ;  $k = 1, 2, \dots, r$  can be determined from the characteristic equation

$$\begin{vmatrix} x_0 & x_1 & \dots & x_m \\ x_1 & x_2 & \dots & x_{m+1} \\ \dots & \dots & \dots & \dots \\ x_{m-1} & x_m & \dots & x_{2m-1} \\ 1 & \rho & \dots & \rho^m \end{vmatrix} = 0; \tag{5}$$

The recurrence indexes of these roots  $n_k$  ( $n_k \in N$ ) satisfy the equality  $n_1 + n_2 + \dots + n_r = m$ ; coefficients  $\mu_{kl} \in C$ ;  $k = 1, 2, \dots, r$ ;  $l = 0, 1, \dots, n_k - 1$  can be determined from a system of linear algebraic equations which can be formed from the systems of equalities in Eq. (4).

The set of characteristic roots  $P(x_0, x_1, \dots, x_{2m-1}) = \{\rho_k \in C; k = 1, 2, \dots, r; n_1 + n_2 + \dots + n_r = m; m \geq 1\}$  is associated to the finite

sequence  $x_0, x_1, \dots, x_{2m-1}$  which is denoted as the base fragment of the algebraic progression [13].

## 3. The forecasting strategy

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

$$x_0, x_1, x_2, \dots, x_{2n-1}; \tag{6}$$

where  $x_{2n-1}$  is the value of the observation at the present moment. Let us assume that the sequence  $(x_k; k \in Z_0)$  is an algebraic progression and its  $H$ -rank is equal to  $n$ . Then it is possible to determine the next element of the sequence  $x_{2n}$  from the following equality:

$$\det^{(n+1)} = \det \begin{bmatrix} x_0 & x_1 & \dots & x_n \\ x_1 & x_2 & \dots & x_{n+1} \\ \dots & \dots & \dots & \dots \\ x_n & x_{n+1} & \dots & x_{2n} \end{bmatrix} = 0; \tag{7}$$

where the only unknown is  $x_{2n}$  (the horizon of the prediction is equal to 1). Unfortunately, real world series are usually contaminated with more or less noise. Thus, such a straightforward assumption that the sequence  $(x_k; k \in Z_0)$  is an algebraic progression does not hold in practice (a random sequence does not have a rank).

### 3.1. The identification of a skeleton algebraic sequence

In the previous paper [13] we have used an evolutionary strategy to identify the algebraic sequence by removing the additive noise. In fact, we were using  $2n+1$  observations to build a model of the process. The idea was based on the assumption that the sequence in Eq. (1) is produced by adding noise to some unknown algebraic progression (the  $H$ -rank of that algebraic progression is assumed to be equal to  $n$ ). In other words, we made a proposition that the sequence

$$\tilde{x}_k = x_k - \varepsilon_k; k = 0, 1, 2, \dots \tag{8}$$

is an algebraic progression and this sequence is some sort of a skeleton sequence determining the global dynamics of the time series. Then, according to Eq. (3),

$$\det(\tilde{H}^{(n+1)}) = 0 \tag{9}$$

where

$$\tilde{H}^{(n+1)} = \begin{bmatrix} \tilde{x}_0 & \tilde{x}_1 & \dots & \tilde{x}_n \\ \tilde{x}_1 & \tilde{x}_2 & \dots & \tilde{x}_{n+1} \\ \dots & \dots & \dots & \dots \\ \tilde{x}_n & \tilde{x}_{n+1} & \dots & \tilde{x}_{2n} \end{bmatrix}. \tag{10}$$

Corrections  $\varepsilon_k$ ;  $k = 0, 1, 2, \dots, 2n$  had to be identified before any predictions could be made. Since the goal was to minimize any distortions of the original time series, the fitness function for the set of corrections  $\{\varepsilon_0, \varepsilon_1, \dots, \varepsilon_{2n}\}$  was introduced in [13]:

$$F(\varepsilon_0, \varepsilon_1, \dots, \varepsilon_{2n}) = \frac{1}{a \left| \det(\tilde{H}^{(n+1)}) \right| + \sum_{k=0}^{2n} \lambda_k |\varepsilon_k|}; a > 0; \tag{11}$$

where

$$\lambda_k = \frac{\exp(b(k+1))}{\sum_{j=0}^{2n} \exp(b(j+1))}; k = 0, 1, \dots, 2n; b > 0. \tag{12}$$

If the original time series is an algebraic progression and  $\det H^{(n+1)} = 0$ , the fitness function reaches its maximum at  $\epsilon_0 = \epsilon_1 = \dots = \epsilon_{2n} = 0$  ( $F(0, 0, \dots, 0) = +\infty$  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  $\lambda_0, \lambda_1, \dots, \lambda_{2n}$  determine the tolerance corridor for corrections  $\epsilon_0, \epsilon_1, \dots, \epsilon_{2n}$ . All corrections would have the same weight if  $b = 0$ . The larger is  $b$ , 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 previous moments. That corresponds to the supposition that the importance of the observation depends on its closeness to the present moment.

It can be observed that such a prediction strategy based on the identification of skeleton algebraic sequences and the fitness function described by Eq. (10) works well for short time series and outperforms many other predictors if a day-ahead local maximum and local minimum must be considered [13]. In general, the variability of the time series forecasted by the described technique is an advantageous factor (though a number of trials are necessary before an averaged estimate of the prediction can be produced).

### 3.2. The proposed scheme

The basic idea of the proposed forecasting scheme can be described by the following considerations. We will try to identify algebraic relationships in the available observation data. But we will try to smooth the forecast before the prediction is done—instead of trying to make a straightforward projection of this algebraic model into the future (as it is done in [13]). In other words, we will try to seek conciliation between the variability of the skeleton algebraic sequences and the smoothness of the averaged estimates.

We will consider the sequence  $x_0, x_1, x_2, \dots, x_{2n-1}$  (Fig. 1). It is clear that a straightforward identification of the next element  $x_{2n}$  using Eq. (9) is not applicable due to the unavoidable additive noise in real world time series (the original sequence is illustrated by a thick solid line in Fig. 1; the straightforward forecast of  $x_{2n}$  is shown by an empty circle).

As mentioned previously, we will try to seek a balance between the straightforward algebraic prediction of the next element of the sequence and the smoothing of current observations. An often used industrial technique to remove inherent random variation

in a collection of data is the simple moving average smoothing (MA) [14]:

$$\bar{x}_k = \frac{1}{s} \sum_{i=0}^{s-1} x_{k-i-1} \quad (13)$$

where  $\bar{x}_k$  is a smoothed value at the moment  $k$ ;  $s$  is the averaging window. In general, the width of the averaging window should be preselected for each time series and is not related to the length of the skeleton algebraic sequence (the averaging window is illustrated by a horizontal arrow in Fig. 1). The smoothed value  $\bar{x}_{2n}$  is shown by a gray-shaded circle in Fig. 1 at  $k = 2n$ .

Let us now construct the fitness function for the set of corrections  $\{\epsilon_0, \epsilon_1, \dots, \epsilon_{2n-1}\}$ :

$$F(\epsilon_0, \epsilon_1, \dots, \epsilon_{2n-1}) = \frac{1}{a \sum_{k=0}^{2n-1} \lambda_k |\epsilon_k| + |\bar{x}_{2n} - x_{2n}|}; \quad (14)$$

where

$$\lambda_k = \frac{\exp(b(k+1))}{\sum_{j=0}^{2n-1} \exp(b(j+1))}; \quad k = 0, 1, \dots, 2n-1; \quad b > 0; \quad (15)$$

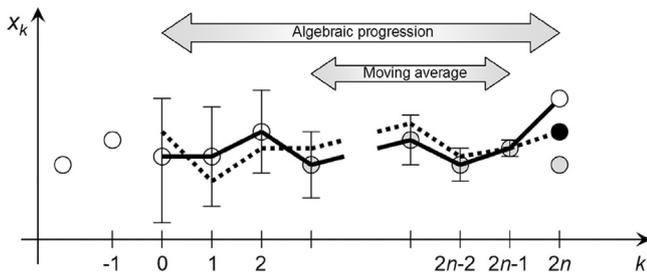
$\tilde{x}_{2n}$  is the solution of Eq. (9);  $\bar{x}_{2n}$  is the smoothed moving average (the result of Eq. (12)) and the parameter  $a > 0$  determines the penalty proportion between the sum of weighted corrections and the difference of forecasts based on skeleton algebraic sequences and moving averages (both penalties have the same weight when  $a = 1$ ). It must be noted that the fitness function does not comprise a term associated to the determinant of the Hankel matrix (compare to Eq. (11)).  $F(0, 0, \dots, 0) = +\infty$  if  $\epsilon_0 = \epsilon_1 = \dots = \epsilon_{2n} = 0$  and the algebraic forecast  $\tilde{x}_{2n}$  is equal to the forecast  $\bar{x}_{2n}$  produced by the MA method. In general, the goal is to maximize the fitness function by making small corrections to the sequence of observations and produce a forecast close to the smoothed moving average. It is clear that  $\sum_{k=0}^{2n-1} \lambda_k = 1$ ;  $0 < \lambda_0 < \lambda_1 < \dots < \lambda_{2n-1} < \lambda_{2n-1}$ ; the penalties for corrections  $\epsilon_0, \epsilon_1, \dots, \epsilon_{2n-1}$  are illustrated by corresponding intervals in Fig. 1. The algebraic sequence  $\tilde{x}_0, \tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_{2n-1}$  is illustrated by a thick dotted line in Fig. 1; the forecast  $\tilde{x}_{2n}$  is shown as a black circle in Fig. 1.

It can be noted that we do not compute an arithmetic average between a straightforward forecast  $x_{2n}$  and the smoothed moving average  $\bar{x}_{2n}$ . As mentioned previously, real-world time series are unavoidably contaminated with more or less noise. Thus the rank of such time series does not exist and the straightforward computation of the forecast  $x_{2n}$  does not have any physical (moreover mathematical) motivation. Instead, our goal is to reconstruct the nearest skeleton algebraic sequence to the original series. Moreover, this skeleton algebraic sequence should produce a forecast close to the smoothed average computed for the last data of the original series. In other words, we construct a scheme of algebraic prediction with the internal smoothing. Particular details about the construction of the computational algorithm and the selection of its parameters will be given in the following sections.

### 3.3. Effects of the additive noise

It is clear that a random sequence does not have an  $H$ -rank (otherwise algebraic relationships governing the evolution of this random sequence could be derived). If the rank of a sequence  $Hr(x_k; k \in Z_0) = m$  and a sequence  $(\epsilon_k; k \in Z_0)$  is a random sequence, then  $Hr(x_k + \epsilon_k; k \in Z_0) = +\infty$  [15]. As mentioned previously, the proposed forecasting method is based on the identification of underlying skeleton algebraic progression in real-world time series contaminated by the inherent noise.

The concept of the pseudospectrum of a square matrix is thoroughly investigated in [16]. Analogous reasoning in regards



**Fig. 1.** The schematic diagram illustrating the proposed method of prediction: circles denote the original time series;  $k = 2n - 1$  is the present moment; the thick solid line denotes a straightforward algebraic prediction according to Eq. (9). (The result of this prediction is illustrated by a white circle at  $k = 2n$ ); the averaging window is denoted by gray-shaded circles (the smoothed prediction is illustrated by a gray-shaded circle at  $k = 2n$ ); vertical intervals denote the tolerance corridor for the corrections of the original time series; the dashed line denotes the corrected skeleton algebraic sequence; the black circle denotes the final prediction.

to pseudo  $H$ -rank could help to understand the effects introduced by the additive noise to the underlying algebraic relationships governing the evolution of sequences (even though the  $H$ -rank of the sequence with the additive noise does not exist).

The spectrum of a square matrix  $A$ , denoted as  $\Lambda(A)$ , is the set of  $z \in \mathbb{C}$  where the resolvent  $(zI - A)^{-1}$  does not exist or is unbounded [16] ( $I$  is the identity matrix). For each  $\varepsilon > 0$ , the  $\varepsilon$ -pseudospectrum of  $A$  is defined by [16]:

$$A_\varepsilon(A) = \{z \in \mathbb{C} : z \in \Lambda(A + E) \text{ for some } E \text{ with } \|E\| \leq \varepsilon\}. \quad (16)$$

In analogy to the classical definition of the spectrum of a square matrix we define the  $H$ -spectrum of the base fragment of the algebraic progression as the set of characteristic roots  $\rho_k$  of the characteristic equation Eq. (5). Then, for each  $\varepsilon > 0$ , the  $\varepsilon$ - $H$ -pseudospectrum is the subset on the complex plane comprising all possible locations of characteristic roots of the perturbed original sequence:

$$P_\varepsilon(x_0, x_1, \dots, x_{2m-1}) = \left\{ \begin{array}{l} z \in \mathbb{C} : z \in P(x_0 + \varepsilon_0, x_1 + \varepsilon_1, \dots, x_{2m-1} + \varepsilon_{2m-1}) \text{ for some } \varepsilon_0, \varepsilon_1, \dots, \varepsilon_{2m-1} \\ \text{with } \|\varepsilon_0, \varepsilon_1, \dots, \varepsilon_{2m-1}\|_2 \leq \varepsilon; \varepsilon_k \in \mathbb{R}; k = 1, 2, \dots, 2m-1 \end{array} \right\} \quad (17)$$

There is a principal difference between Eqs. (15) and (16)—we do perturb not the Hankel matrix but the elements of the base fragment of the algebraic progression instead (our motivation can be explained by the fact what we do explore the extrapolation of the sequence). Moreover, we avoid the conflict associated to the nonexistence of the  $H$ -rank of the algebraic progression contaminated with additive noise. We do not define (and do not perturb) the element  $x_{2m}$ . This element can be solved from the equation  $d^{(m+1)} = 0$  (Eq. (3)). Thus the  $H$ -rank of the perturbed sequence remains equal to the  $H$ -rank of the unperturbed sequence. Another difference is based on the fact that the perturbing matrix  $E$  in the classical definition of the pseudospectrum comprises complex numbers while we employ the perturbing vector comprising real numbers only. As mentioned previously, this can be explained by the fact that we extrapolate real time series only.

The computation of the  $\varepsilon$ - $H$ -pseudospectrum requires finding roots of the perturbed characteristic equation (Eq. (5)). The  $m$  roots of the polynomial of degree  $m$  depend continuously on the coefficients [17] (though the problem of approximating the roots given the coefficients is ill-conditioned [18]). The coefficients of the polynomial are appropriate adjuncts of the determinant in Eq. (5). But

$$\det(A + \varepsilon E) - \det(A) = \det(A) \operatorname{tr}(A^{-1} E) \varepsilon + O(\varepsilon^2) \quad (18)$$

Thus, following properties hold for a small perturbation of the base fragment of the algebraic progression in the described computational setup:

- (i) it does not change the  $H$ -rank of the sequence;
- (ii) the  $\varepsilon$ - $H$ -pseudospectrum converges continuously to the  $H$ -pseudospectrum as  $\varepsilon \rightarrow 0$ ;
- (iii) all roots of the perturbed characteristic polynomial are either real numbers or complex conjugate numbers because all elements of the perturbed base fragment of the algebraic progression are real.

In other words, the removal of corrections  $\varepsilon_0, \varepsilon_1, \dots, \varepsilon_{2m-1}$  from the real-world time series (although based on the evolutionary strategy) is a well-posed problem.

**Example 1.** A simple computational example is used to illustrate the concept of the  $\varepsilon$ - $H$ -pseudospectrum. Let us consider a periodic sequence  $\{-1, 1, 2, -1, 1, 2, \dots\}$  with the period equal to 3 ( $m = 3$ ). Then, elementary computations yield characteristic roots

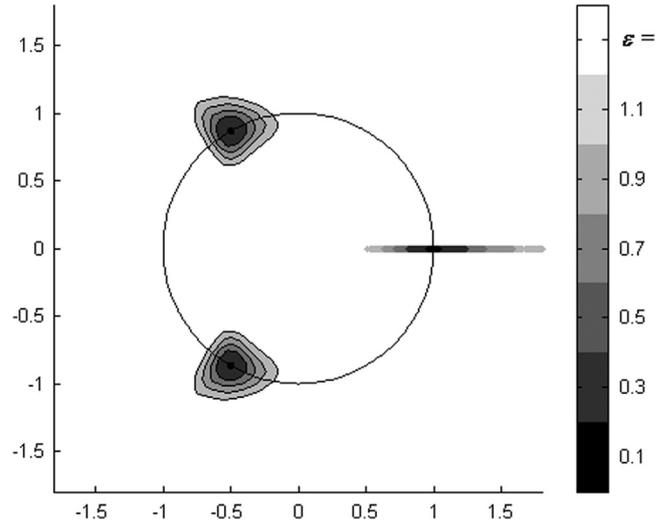


Fig. 2. The  $\varepsilon$ - $H$ -pseudospectrum of a periodic sequence  $\{-1, 1, 2, -1, 1, 2, \dots\}$ . Smooth convergence to the  $H$ -pseudospectrum is observed as  $\varepsilon$  tends to 0.

for this sequence:  $\rho_1 = 1$ ;  $\rho_2 = -1/2 + \sqrt{3}/2i$  and  $\rho_3 = -1/2 - \sqrt{3}/2i$  (all roots for a periodic sequence are located on the unit circle in the complex plane—see the formal proof in [13]). Now we fix the constant  $\varepsilon$  and construct 1000 vectors of corrections  $[\varepsilon_0, \varepsilon_1, \dots, \varepsilon_5]$  such that  $\|\varepsilon_0, \varepsilon_1, \dots, \varepsilon_5\|_2 = \varepsilon$ . A random number generator is used for the construction of such vectors:  $\varepsilon_k = \varepsilon / \|\varepsilon_0, \varepsilon_1, \dots, \varepsilon_5\|_2$  where  $\varepsilon_k; k = \overline{0, 5}$  are random numbers distributed uniformly in the interval  $[-1; 1]$ . Characteristic roots for the perturbed sequence are calculated for every correction and plotted in the complex plane. The contour plotter and different grayscale levels are used to illustrate different regions of the  $\varepsilon$ - $H$ -pseudospectrum (Fig. 2).

It is interesting to note that the first root of the perturbed sequence remains real while the other two roots are still complex conjugate (the necessary condition for the perturbed sequence to remain real). The perturbed sequence is no longer periodic, but the  $\varepsilon$ - $H$ -pseudospectrum converges continuously to the  $H$ -pseudospectrum as  $\varepsilon \rightarrow 0$ .

### 3.4. A simple numerical example

Let us illustrate the concept of the proposed forecasting scheme by a simple numerical example. Let four observations are available:  $x_0 = 1$ ;  $x_1 = 2$ ;  $x_2 = 0$  and  $x_3 = 2$ . Let the averaging window  $s = 2$ ; then the smoothed prediction is  $\bar{x}_4 = 1$ . The straightforward

algebraic forecast is  $x_4 = -1$  because  $\begin{vmatrix} 1 & 2 & 0 \\ 2 & 0 & 2 \\ 0 & 2 & -1 \end{vmatrix} = 0$ . For the

simplicity we will assume that  $a = 1$  and  $b = 0$ . Then we need to find such corrections  $\varepsilon_k; k = 0, 1, 2, 3$  that maximize the value of the fitness function defined by Eq. (13). Eq. (9) yields the value of the algebraic forecast:

$$\bar{x}_4 = \frac{-\varepsilon_2((2 + \varepsilon_1)(2 + \varepsilon_3) - \varepsilon_2^2) + (2 + \varepsilon_3)((1 + \varepsilon_0)(2 + \varepsilon_3) - \varepsilon_2(2 + \varepsilon_1))}{(1 + \varepsilon_0)\varepsilon_2 - (2 + \varepsilon_1)^2} \quad (19)$$

We need to determine such values of  $\varepsilon_k; k = 0, 1, 2, 3$  that the value of the fitness function  $F(\varepsilon_0, \varepsilon_1, \varepsilon_2, \varepsilon_3)$  is optimal. The lowest bound of the fitness function is  $F(0, 0, \dots, 0) = 1/|\bar{x}_{2n} - \bar{x}_{2n}| = 0.5$ . But the selection of the optimal corrections  $\varepsilon_k; k = 0, 1, 2, 3$  is not a trivial task even for this simple example. The situation would become much more complex in case of realistic prediction

scenarios. Therefore the development of a reliable and an efficient optimization strategy becomes a subject of the primary importance for the successful implementation of such a forecasting strategy.

#### 4. Parameter selection in PSO

It is clear that the prediction of an algebraic sequence by the proposed algebraic forecasting method with internal smoothing cannot be exact. The direct computation of the “forecast” using Eq. (7) is of course analytic (i.e. exact). But the fitness function in Eq. (13) does not comprise a term representing the determinant of the Hankel matrix. In other words, the exact forecast of an exact algebraic sequence is impossible because the value of the forecast can be far from the smoothed average (note that the exact prediction of an algebraic sequence works well with the fitness function in Eq. (11) [13]). As mentioned previously, there does not exist one method which would outperform all others, in all situations. It is natural to expect that the proposed method should work well with such signals where the noise–signal ratio is rather high.

We will use an artificial test time series to tune parameters of the proposed forecasting algorithm. We form a periodic sequence (numerical values of seven elements in a period are selected as 0.5; 0.7; 0.1; 0.9; 0.3; 0.2; 0.8); this sequence represents a skeleton algebraic sequence. We add random numbers uniformly distributed in the interval  $[-0.15; 0.15]$  to all elements of that sequence. This test time series will be used for testing the functionality of the proposed method.

The first task is to identify the  $H$ -rank of the time series (incorrect identification of an appropriate  $H$ -rank may lead to substantial prediction errors). We use Eq. (7) to solve  $\tilde{x}_{2n}$  without using any corrections or moving averages ( $\tilde{x}_{2n}$  does not necessarily coincide with  $x_{2n}$ ). Then we shift the observation window by one element forward and again use Eq. (7) to solve the next element  $\tilde{x}_{2n+1}$ . Such direct one-step forward forecasting is repeated for 50 times; root mean square errors (RMSE) of such direct algebraic prediction are shown in Table 1. Best results are produced at  $m = 7$  (the dimension of the characteristic Hankel matrix is 8), thus we assume that the  $H$ -rank of the test time series is 7.

The averaging window  $s$  for the simple moving average smoothing (Eq. (12)) and the penalty proportion parameter  $a$

(Eq. (13)) must be selected in the next step (the parameter  $b$  is set to 0). We will use the test time series again, but evolutionary algorithms will be used now to identify the near-optimal set of corrections  $\{\varepsilon_0, \varepsilon_1, \dots, \varepsilon_{13}\}$ .

Particle swarm optimization (PSO) techniques have been successfully employed in [13] for the identification of the skeleton algebraic sequence. We will also use PSO for the selection of a near-optimal set of corrections. And though despite numerous research efforts the selection of the parameters of PSO remains mostly empirical and depends on the topology of the target function and/or on the structure of the fitness function [19,20], we fix  $w = 0.6$  and  $c_1 = c_2 = 1.7$  as recommended by Trelea [21] ( $c_1$  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*;  $w$  is the inertia weight balancing the global and the local search). There have been no definitive recommendations in the literature regarding the swarm size in PSO. Eberhart and Shi [22] 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 to 60, but there are no established guidelines [23]. For the purpose of comparing the efficiency of the predictor developed in [13] and the proposed method, the swarm size that we use for PSO is fixed to 50 particles.

It is clear that a new set of near-optimal corrections  $\{\varepsilon_0, \varepsilon_1, \dots, \varepsilon_{2m-1}\}$  is generated every time when the PSO algorithm is executed. Thus we execute PSO algorithm 100 times, compute the forecasted value of  $\tilde{x}_{2m}$  (100 different estimates of  $\tilde{x}_{2m}$  are produced in the process) and calculate root mean square errors (RMSE) between the true value of  $x_{2m}$  and 100 forecasted estimates of  $\tilde{x}_{2m}$ . Moreover, we shift the observation window by one step forward and repeat 100 trials again. Such procedures are repeated 50 times, all RSME estimates are arithmetically averaged. Results of computational experiments are presented in Table 2.

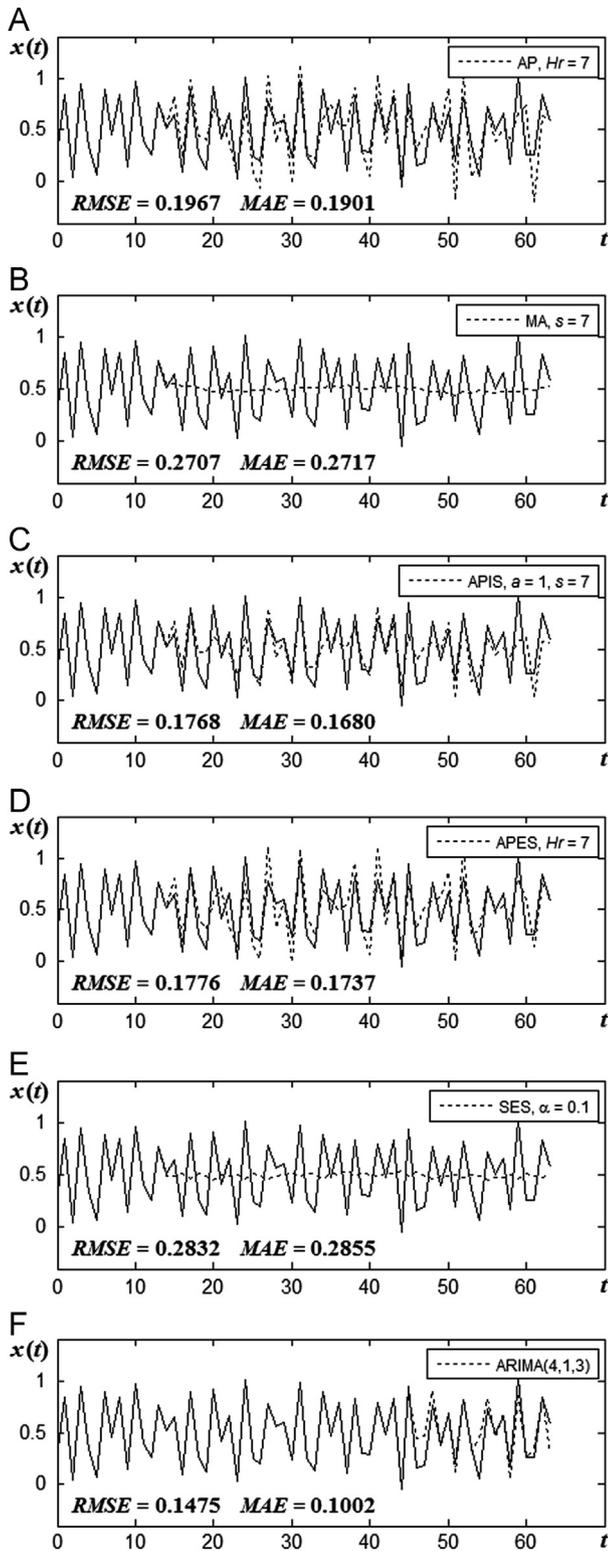
As mentioned previously, such computational experiments are performed at different  $s$  (the time averaging window) and  $a$  (the penalty proportion parameter). We select such discreet values of the parameter  $a$ :  $1/4m$ ;  $1/2m$ ;  $1/m$ ;  $1/2$ ;  $1$ ;  $2$ ;  $m/2$ ;  $m$ ;  $2m$  and  $4m$  ( $m = 7$  for the test time series), while  $s = 2, 3, \dots, 2m$ . Note that the index of  $\varepsilon_k$  runs from 0 to  $2m - 1$ ; thus the maximum number of available elements for the simple moving average smoothing is  $2m$ . The best prediction result (RMSE = 0.1768) is produced at  $a = 1$

**Table 1**  
RMSE of the direct algebraic prediction for the test time series at different  $m$ .

$m$	4	5	6	7	8	9	10	11	12	13	14
RMSE	1.2960	2.1512	21.5646	<b>0.1967</b>	28.7332	1.0214	3.4570	18.4699	7.2924	18.4942	1.3620

**Table 2**  
RMSE of the algebraic prediction for the test time series with internal smoothing at different  $s$  (the time averaging window) and  $a$  (the penalty proportion parameter);  $m = 7$ .

$s/a$	$1/4m$	$1/2m$	$1/m$	$1/2$	1	2	$m/2$	$m$	$2m$	$4m$
2	0.2222	0.2257	0.2168	0.1948	0.1953	0.1933	0.1948	0.1960	0.1962	0.2025
3	0.2209	0.2197	0.2126	0.1940	0.1914	0.1908	0.1940	0.1982	0.2022	0.2041
4	0.2113	0.2157	0.2114	0.1912	0.1893	0.1911	0.1912	0.1952	0.1992	0.2033
5	0.2079	0.2047	0.2123	0.1930	0.1840	0.1871	0.1930	0.1962	0.1963	0.2052
6	0.2100	0.2127	0.2124	0.1963	0.1866	0.1915	0.1963	0.1986	0.2026	0.2012
7	0.2004	0.1995	0.1971	0.1951	<b>0.1768</b>	0.1833	0.1951	0.1978	0.1975	0.2011
8	0.2120	0.2056	0.2055	0.1940	0.1822	0.1879	0.1940	0.1987	0.1976	0.2053
9	0.2108	0.2077	0.200	0.1930	0.1825	0.1862	0.1930	0.2006	0.2046	0.2045
10	0.2018	0.1977	0.2063	0.1964	0.1812	0.1887	0.1964	0.1965	0.2005	0.2002
11	0.2038	0.2058	0.2083	0.1966	0.1809	0.1886	0.1966	0.1963	0.2003	0.2052
12	0.1966	0.2043	0.2013	0.1951	0.1804	0.1864	0.1951	0.1945	0.1951	0.2016
13	0.2059	0.2017	0.2043	0.1945	0.1828	0.1847	0.1945	0.1969	0.2009	0.2028
14	0.1990	0.1977	0.1959	0.1960	0.1791	0.1857	0.1960	0.1951	0.1974	0.2006



**Fig. 3.** Forecasts of the test time series by the direct algebraic predictor (A); the MA method (B); the algebraic predictor with internal smoothing (C); the algebraic predictor with external smoothing (D); the SES method (E) and the ARIMA method (F).

and  $s = 7$  (Table 2). Thus we fix these values of parameters ( $a = 1$  and  $s = m$ ) and will use them for the prediction of other time series (the  $H$ -rank  $m$  must be identified for each individual time series before any predictions could be commenced). The selection of the optimal value of parameter  $b$  is extensively discussed in [13]; we adopt the near-optimal value  $b = 1$  in APIS scheme too.

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

#### A. Preprocessing.

- (1) Identify the  $H$ -rank of the time series (the parameter  $m$ ) by performing direct algebraic predictions (without using any corrections or moving averages) for different  $m$ . The smallest RMSE of the direct algebraic prediction is used for the selection of the optimal  $m$ .
- (2) Set the penalty proportion parameter  $a = 1$  and the averaging window for the simple moving average smoothing  $s = m$ .
- (3) Set the inertia weight  $w = 0.6$  and the acceleration constants  $c_1 = c_2 = 1.7$  for the PSO algorithm.

#### B. One-step forward prediction algorithm.

- (1) Compute the smoothed moving average  $\bar{x}_{2m}$  from  $\{x_m, x_{m+1}, \dots, x_{2m-1}\}$ .
- (2) Repeat 100 times:
  - (2.1) Compute a single set of corrections  $\{\epsilon_0, \epsilon_1, \dots, \epsilon_{2m-1}\}$  using the PSO fitness function (Eq. (13)).
  - (2.2) Fix the algebraic forecast with internal smoothing  $\tilde{x}_{2m}$ .
- (3) Compute the averaged forecast of  $\tilde{x}_{2m}$ .
- (4) Shift the observation window by 1 step forward and return to step (B.1).

## 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 the proposed forecasting technique with other methods. Direct algebraic prediction (without internal smoothing) is illustrated in Fig. 3(A); RMSE of such a straightforward prediction is 0.1967. The prediction performed by the moving average (MA) method (at  $s = 7$ ) produces a higher RMSE and demonstrates explicit averaging features of the method (Fig. 3(B)). Direct algebraic forecasting already outperforms MA if RMSE metrics would be considered only. But the variability of the predicted time series is incomparably better in Fig. 3(A) than in Fig. 3(B) (though some overestimates of local minimums and local maximums can be observed in Fig. 3(A)). Algebraic prediction with internal smoothing (APIS) produces the best RMSE and the best estimates of local extremes for the test time series (Fig. 3(C)). Local fluctuations of the predicted time series by our method give a much better representation of the variability of the time series. For instance, our method would clearly outperform the MA method if one would be interested to identify a day-ahead local maximum and local minimum [2]. The conciliation of powerful algebraic variability and the smoothness of moving averaging helps to unleash the power of the proposed technique.

A prediction method based on the identification of algebraic skeleton sequences is developed in [13]. The one-step-forward forecast of this method is constructed as an arithmetic average of successful trials—thus this prediction method can be denoted as the algebraic prediction with external smoothing (APES) in Fig. 3(D). By the way we use the same test series as in [13]; RMSE of APES prediction is slightly higher compared to APIS prediction. As mentioned previously, RMSE is the only one indicator describing the quality of the predicted time series. APES produces worse estimates of the day-ahead local maximums and local minimums compared to the APIS forecast (Fig. 3).

We also compare the functionality of APIS with the predictor based on sequential exponential smoothing (SES) [24] which is an often used industrial technique to remove inherent random variation in a collection of data. It is a simple and pragmatic approach to forecasting, whereby the forecast is constructed from an exponentially weighted average of past observations [24]. Single exponential smoothing (SES) method assigns exponentially decreasing weights as the observation get older:

$$S_t = \alpha x_{t-1} + (1 - \alpha)S_{t-1} \quad (20)$$

where  $t = 1, 2, 3, \dots$ ;  $S_1 = x_0$  and  $\alpha$  is smoothing factor;  $0 \leq \alpha \leq 1$  [25,26]. Though the selection of a best value of the parameter  $\alpha$  is discussed in [24], we perform a series of computational experiments and identify the best value (in terms of RMSE) for the test time series –best results are produced at  $\alpha = 0.1$ . It can be noted that SES may not excel if a trend or seasonality is present in data; double or even triple exponential smoothing may produce better results then [25]. The test series does not contain a clearly expressed trend or seasonality, therefore we run computational experiments with a SES only. The produced RMSE is 0.2832; the results are presented in Fig. 3(E).

We continue computational experiments with Box-Jenkins's time series autoregressive integrated moving average procedure ARIMA(4,1,3) [27] (Fig. 3(F)) as the experiments found the 4–1–3 architecture as the best model for this 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). The produced RMSE of the ARIMA forecast is 0.1475 and outperforms APIS forecast (Fig. 3). Nevertheless, it can be observed that APIS method produces the first forecast from 14 available elements of the original sequence ( $m=7$ ), whereas ARIMA requires a considerably longer data sequence before statistical parameters can be identified and the forecasting can be commenced. APIS method is based on the reconstruction of near-optimal algebraic skeleton sequences; we do reconstruct 100 different skeletons from 14 available elements. And though the computational complexity of such an approach increases, the proposed method is a good candidate for the prediction of very short-term data sequences.

Note that we preselect the individual architecture of the ARIMA model for each time series, while APIS parameters  $a$  and  $b$  are tuned for the artificial time series and a kept fixed for all other time series. It is likely that APIS prediction results would be even better if parameters  $a$  and  $b$  would be also individually tuned for each time series. But we avoid such an individual tuning simply because the complexity of such forecasting strategy would increase considerably compared to the algorithm proposed at the end of Section 4.

## 5.2. The test time series with the Gaussian noise

Computational experiments are repeated with the test series, but we add Gaussian noise (zero mean and standard variance equal to 0.1) instead of uniform noise to the same periodic sequence. Direct algebraic prediction (without internal smoothing) is illustrated in Fig. 4(A); RMSE of such a straightforward prediction is 0.2122. The prediction performed by the moving average (MA) method (at  $s = 7$ ) produces a slightly higher RMSE (Fig. 4(B)). APIS predictor demonstrates low RMSE and good variability (Fig. 4(C)). RMSE of APES predictor is almost as good as the one produced by APIS, but the over-estimation of local minimums and maximums can be observed in Fig. 4(D). SES produces best results at  $\alpha = 0.1$  (Fig. 4(E)).

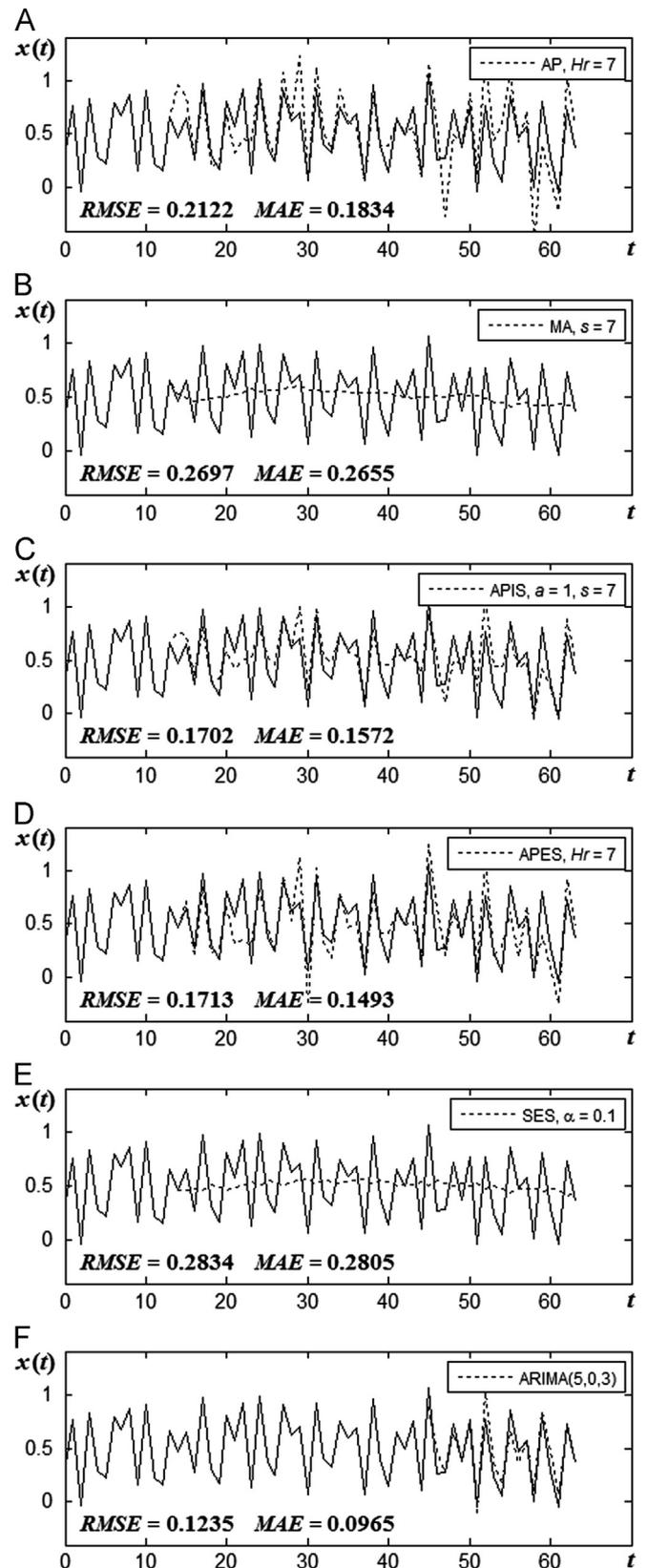


Fig. 4. Forecasts of the test time series with the Gaussian noise by the direct algebraic predictor (A); the MA method (B); the APIS method (C); the APES method (D); the SES method (E) and the ARIMA method (F).

## 5.3. Andrews46.dat time series

We continue computational experiments with real-world time series. The functionality of the proposed APIS predictor is tested

using Andrews46.dat time series representing the annual yield of straw on Broadbalk field at Rothamsted in the period of 1852–1925 [28]. Andrews46.dat time series comprises 74 positive real elements; we norm this series by dividing all elements by the maximum element in this sequence.

The first task is to identify the length of the base fragment of Andrews46.dat time series. Direct algebraic prediction yields lowest RMSE at  $m=2$  (Fig. 5(A)); this value is set for further analysis. As mentioned previously, we fix  $s = m = 2$  and perform MA prediction (Fig. 5(B)). APIS and APES forecasts are shown in Fig. 5(C and D); SES forecast (the lowest RMSE is achieved at  $\alpha = 0.1$ ) is shown in Fig. 5(E).

APIS can be also compared with the naïve method (MA at  $s = 1$ ); the resulting RMSE of the naïve method is 0.4068 (MAE is 0.3746). The naïve extrapolation cannot compete with APIS (prediction errors are almost double). In general, APIS outperforms MA methods (including  $s = 1$ ) for Andrews46.dat time series. The main advantage of APIS is a realistic variability of the predicted time series, whereas the time series is essentially smoothed to its current trend by MA at  $s \geq 2$ .

#### 5.4. Odonovan1.dat time series

Odonovan1.dat time series represents consecutive yields of batch chemical processes [28]. All 70 elements of this series are normed by dividing all elements by the maximum element in this sequence. The first task is to identify the length of the base fragment of Odonovan1.dat time series. Direct algebraic prediction yields lowest RMSE at  $m=3$  (Fig. 6(A)); this value is set for further analysis. We fix  $s = 3$  and perform MA prediction (Fig. 6(B)). APIS and APES forecasts are shown in Fig. 6(C and D); SES forecast (the lowest RMSE is achieved at  $\alpha = 0.1$ ) is shown in Fig. 6(E).

#### 5.5. Montgome8.dat time series

Montgome8.dat time series [28] comprises 100 positive elements which are normed by dividing all elements by the maximum element in this sequence. Direct algebraic prediction yields lowest RMSE at  $m=9$  (Fig. 7(A)); this value is set as the  $H$ -rank of this sequence. We fix  $s = 9$  and perform MA prediction (Fig. 7(B)). APIS and APES forecasts are shown in Fig. 7(C and D); SES forecast (the lowest RMSE is achieved at  $\alpha = 0.1$ ) is shown in Fig. 7(E).

As mentioned previously, no single method will outperform all other methods in all situations. The best forecast (in RMSE metrics) is demonstrated by the MA predictor and though APIS yields a worse forecast than MA, it is incomparably better than APES. In other words, the proposed APIS method opens new possibilities for algebraic prediction of real-world time series.

#### 5.6. The Spanish market of electricity price, the winter week from February 18 to February 24, 2002

APIS forecasts of prices in the electricity market of mainland Spain during the winter week from February 18 to February 24, 2002 [29] are presented in Fig. 8. The functionality of the proposed algorithm was compared with different time series forecasting methods, but not with nonlinear forecasting methods such as neural networks (NN) or support vector machines (SVM).

The electricity market time series of mainland Spain has been used to test the forecasting algorithm based on NN [29]. The method based on NN produces  $\sqrt{SSE}=37.92$ ; APIS method results into  $\sqrt{SSE}=92.58$ . The method based of neural networks clearly outperforms APIS. But neural networks need a considerable amount of data to train the network before the prediction can be commenced. The method proposed in [29] uses hourly data on electricity prices during previous 42 days (1008 data points) to

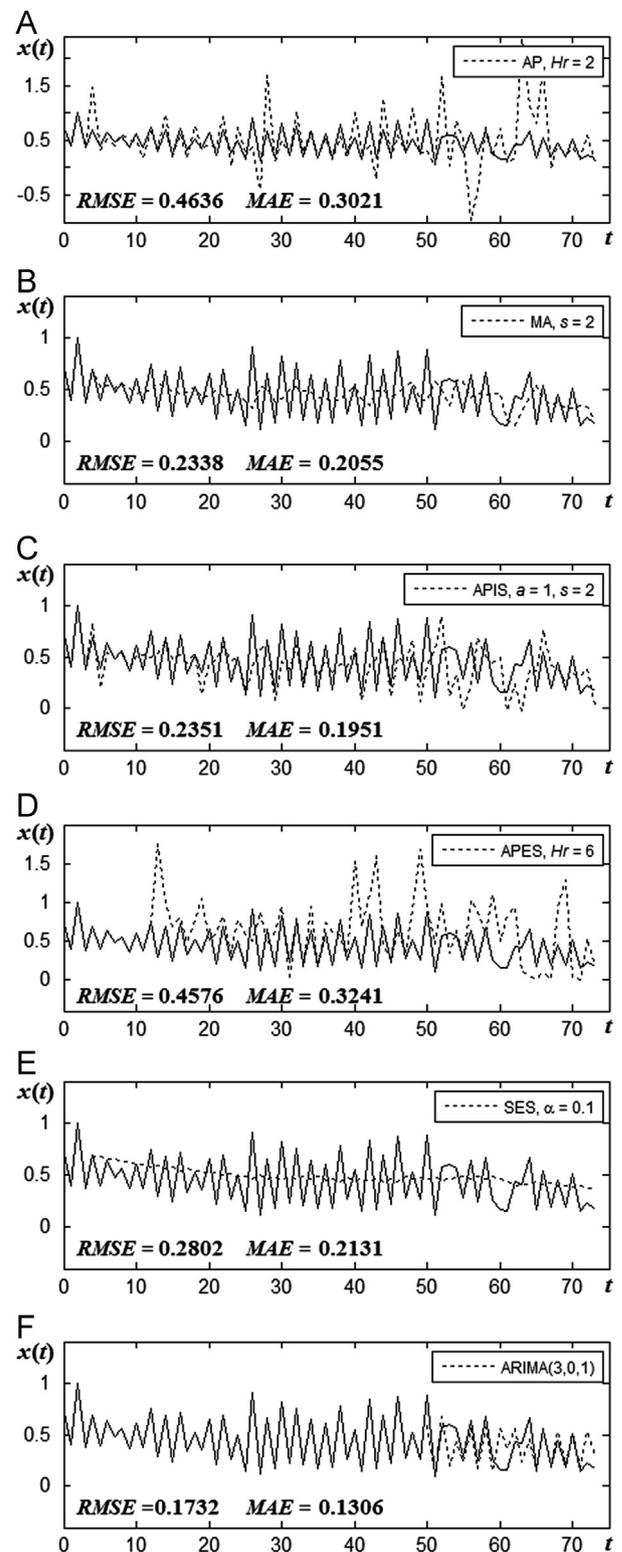


Fig. 5. Forecasts of Andrews46.dat time series by the direct algebraic predictor (A); the MA method (B); the APIS method (C); the APES method (D); the SES method (E) and the ARIMA method (F).

train the network. APIS uses only 48 data points to produce the forecast. But the Spanish electricity time series is a long time series (no shortage of data exists in the field of load forecasting) and the inferior model (APIS) with a small amount of data cannot be considered as an advantage for this time series.

A similar situation occurs with predictions based on SVM. The length of the Mackey–Glass time series used only for the training

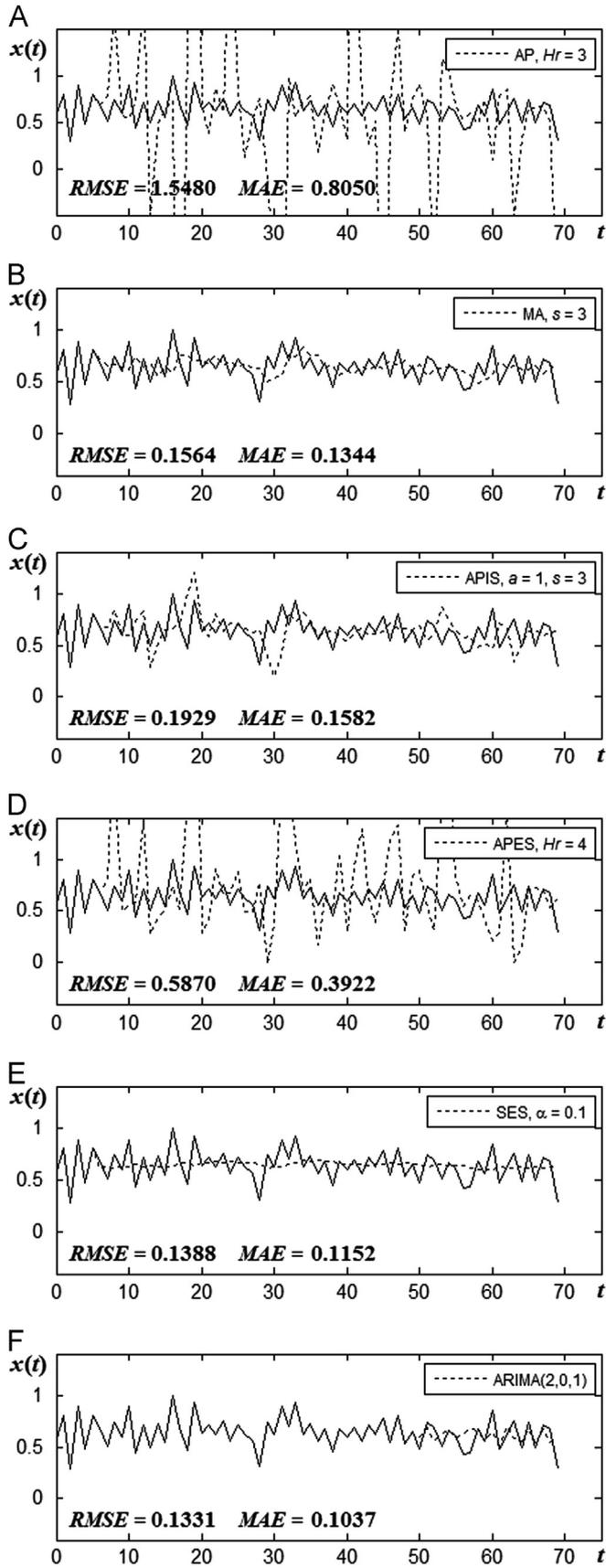


Fig. 6. Forecasts of Odonovan1.dat time series by the direct algebraic predictor (A); the MA method (B); the APIS method (C); the APES method (D); the SES method (E) and the ARIMA method (F).

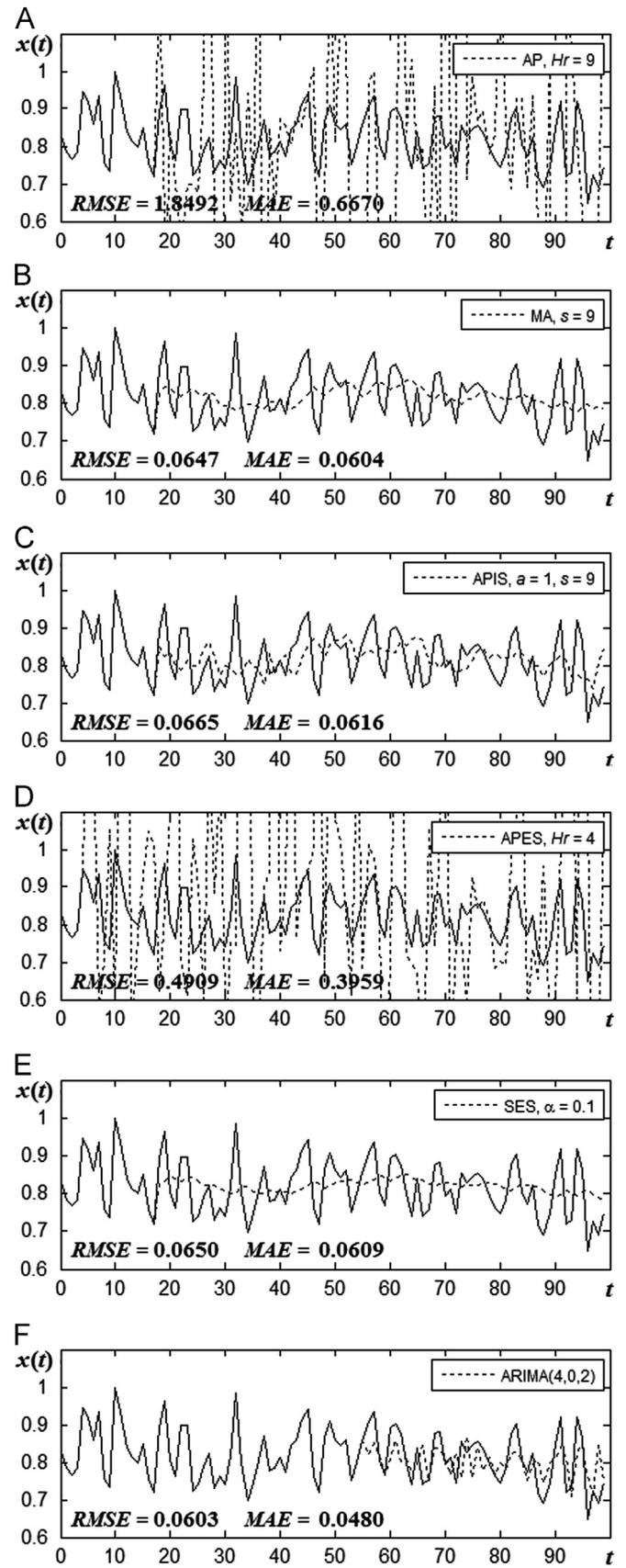


Fig. 7. Forecasts of Montgome8.dat time series by the direct algebraic predictor (A); the MA method (B); the APIS method (C); the APES method (D); the SES method (E) and the ARIMA method (F).

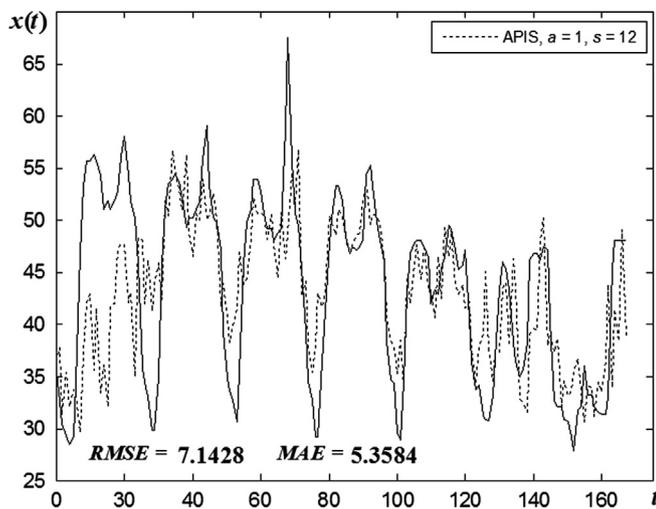


Fig. 8. APIS forecasts of prices in the electricity market of mainland Spain during the winter week from February 18 to February 24, 2002.

Table 3

RMSE and MAE of ten consecutive APIS predictions for the Spanish market of electricity price.

Experiment no	RMSE	MAE
1	7.2996	5.5124
2	7.0912	5.3015
3	7.0532	5.2825
4	6.8669	5.0416
5	6.9628	5.2171
6	7.3346	5.5426
7	7.2386	5.4399
8	7.4367	5.6624
9	7.0912	5.3015
10	7.0532	5.2825
Average	7.1428	5.3584
St. Dev.	0.1785	0.1811
Median	7.0912	5.3015

of SVM is 2000; the NRMS error of the prediction is 0.0159 [30]. Clearly, the Mackey–Glass time series is a chaotic solution of a nonlinear differential equation and one can produce a sequence as long as required. The ANFIS (Adaptive Network-based Fuzzy Inference System) predictor with non-uniform attractor embedding can be used for the forecasting of the Mackey–Glass time series instead; the RMSE of prediction errors is 0.0003497 then [31]. Again, APIS cannot be considered advantageous over SVM for the Mackey–Glass time series.

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. APIS could be considered in such forecasting applications where data scarcity is a definite constraint. A typical example could be gene expression data time series from microarray experiments [32]; such time series usually comprises 10–15 time points (or even fewer). Each temporal gene expression profile is individual—no associations with previous experiments can be made. In other words, offline training is not possible simply because there are no more data available. APIS could be used for one step-forward prediction of such a short time series, while NN and SVM—not.

The proposed forecasting method is based on the identification of a near-optimal set of corrections, the reconstruction of the algebraic model of the time series and the extrapolation this model into the future. The identification of corrections is performed using PSO algorithms. A single run may not guarantee the

repeatability and reproducibility of results since PSO is a stochastic optimization method. We try to avoid random deflections and average 100 trials for the same data set. Such averaging enables to smooth random deflections and helps to achieve the reproducibility of results. In order to test this reproducibility we repeat 10 computational experiments with the electricity price time series; RMSE and MAE error indexes for each experiment are listed in Table 3. The mean, median and variance values are 7.1428, 7.0912, 0.1785 for RMSE and 5.3584, 5.3015, 0.1811 for MAE. It is clear that the reproducibility of the proposed method is rather good.

## 6. Concluding remarks

A method for a short-term time series algebraic forecasting with internal smoothing is proposed in this paper. It is based on the identification of skeleton algebraic sequences and finds a near-optimal balance between algebraic variability and the smoothness of moving averages. The proposed method is especially effective when the time series is short and there are not sufficient data to train models based on neural or fuzzy networks.

So far, we have tuned the parameters of the proposed forecasting algorithm (the penalty proportion between the magnitude of the determinant and the sum of weighted corrections  $a$  and the averaging window  $s$ ) based on the computational experiments with the artificial time series; these values of parameters were used to forecast real-world time series also. One could expect even better results if the parameters  $a$  and  $s$  would be individually tuned for every time series.

On the other hand, it is quite probable that the forecasting accuracy of the proposed method can be improved by the introduction of variable lengths of base fragments at different locations of the forecasted time series. Such computational procedure is directly related to the segmentation of the time series and such adaptive identification of the skeleton algebraic sequences remains a definite target of the future research.

## References

- [1] W.R. Christiaanse, Short term load forecasting using general exponential smoothing, *IEEE Trans. Power Apparatus Syst.* 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-organising fuzzy neural network, *Eng. Appl. Artif. Intell.* 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, *Autom. Electr. Power Syst.* 31 (11) (2007) 31–34.
- [4] V. Vahidinasab, S. Jadid, A. Kazemi, Day-ahead price forecasting in restructured power systems using artificial neural networks, *Electr. Power Syst. Res.* 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, *Util. 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, *Appl. Soft Comput.* 7 (2) (2007) 569–576.
- [7] J. Xue, Z. Shi, Short-time traffic flow prediction based on chaos time series theory, *J. Transp. Syst. Eng. Inf. Technol.* 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 Trans. Power Syst.* 24 (1) (2009) 20–27.
- [10] S. Casolari, M. Colajanni, Short-term prediction models for server management in Internet-based contexts, *Decis. Support Syst.* 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, *J. Central South Univ. Technol.* 15 (s2) (2009) 316–320.
- [12] Da Liu Dongxiao Niu, Wu Desheng Dash, A soft computing system for a day-ahead electricity price forecasting, *Appl. Soft Comput.* 10 (3) (2010) 868–875.

- [13] M. Ragulskis, K. Lukoseviciute, Z. Navickas, R. Palivonaite, Short-term time series forecasting based on the identification of skeleton algebraic sequences, *Neurocomputing* 74 (10) (2011) 1735–1747.
- [14] <http://lorien.ncl.ac.uk/ming/filter/filewma.htm>.
- [15] L.N. Trefethen, Pseudospectra of linear operators, *SIAM Rev.* 39 (1997) 383–406.
- [16] L.N. Trefethen, Computation of pseudospectra, *Acta Numerica* 8 (1999) 247–295.
- [17] E.E. Tyrtshnikov, *A Brief, Introduction to Numerical Analysis*, Birkhauser, Boston, 1997.
- [18] J.H. Wilkinson, The evaluation of the zeros of ill-conditioned polynomials Part I, *Numerische Mathematik* 1 (1959) 150–166.
- [19] 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) 81–86.
- [20] 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) 1951–1957.
- [21] I.C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, *Inf. Process. Lett.* 85 (6) (2003) 317–325.
- [22] 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) 591–600.
- [23] J. Kennedy, R.C. Eberhart, Y.H. Shi, *Swarm Intelligence*, Morgan Kaufman, San Francisco, CA, 2001.
- [24] J.W. Taylor, Smooth transition exponential smoothing, *J. Forecasting* 23 (2004) 385–394.
- [25] <http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc42.htm>.
- [26] 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.
- [27] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, *Time Series Analysis: Forecasting and Control*, Prentice-Hall, Englewood Clis, NJ, 1994.
- [28] R. J. Hyndman, *Time Series Data Library*, <http://datamarket.com/data/list?q=provider:tsdl>.
- [29] J.P.S. Catalao, S.J.P.S. Mariano, V.M.F. Mendes, L.A.F.M. Ferreira, Short-term electricity prices forecasting in a competitive market: a neural network approach, *Electr. Power Syst. Res.* 77 (2007) 1297–1304.
- [30] Zhiwei Shi, Min Han, Support vector echo-state machine for chaotic time series prediction, *IEEE Trans. Neural Networks* 18 (2) (2007) 359–372.
- [31] 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.
- [32] J. Ernst, G.J. Nau, Z. Bar-Joseph, Clustering short time series gene expression data, *Bioinformatics* 21 (Suppl. 1) (2005) i159–i168.



**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.



**Minvydas Ragulskis** 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>).