Physics Letters A 380 (2016) 696–706

Contents lists available at ScienceDirect

Physics Letters A

www.elsevier.com/locate/pla

NARX prediction of some rare chaotic flows: Recurrent fuzzy functions approach

Sobhan Goudarzi [a], Sajad Jafari [a,*], Mohammad Hassan Moradi [a], J.C. Sprott [b]

[a] Biomedical Engineering Department, Amirkabir University of Technology, Tehran 15875-4413, Iran
[b] Department of Physics, University of Wisconsin–Madison, Madison, WI 53706, USA

ARTICLE INFO

Article history:
Received 24 September 2015
Received in revised form 26 November 2015
Accepted 27 November 2015
Available online 30 November 2015
Communicated by C.R. Doering

Keywords:
Rare chaotic flows
Hidden attractor
Fuzzy functions
Recurrent structure
Long term prediction

ABSTRACT

The nonlinear and dynamic accommodating capability of time domain models makes them a useful representation of chaotic time series for analysis, modeling and prediction. This paper is devoted to the modeling and prediction of chaotic time series with hidden attractors using a nonlinear autoregressive model with exogenous inputs (NARX) based on a novel recurrent fuzzy functions (RFFs) approach. Case studies of recently introduced chaotic systems with hidden attractors plus classical chaotic systems demonstrate that the proposed modeling methodology exhibits better prediction performance from different viewpoints (short term and long term) compared to some other existing methods.

1. Introduction

Most familiar examples of low-dimensional chaotic flows occur in systems having one or more saddle points. Such saddle points allow homoclinic and heteroclinic orbits and the prospect of rigorously proving the chaos when the Shilnikov condition is satisfied. Furthermore, such saddle points provide a means for locating any strange attractors by choosing an initial condition on the unstable manifold in the vicinity of the saddle point. Such attractors have been called “self-excited,” and they are overwhelmingly the most common type described in the literature.

Recently, many new chaotic flows have been discovered that are not associated with a saddle point, including ones without any equilibrium points, with only stable equilibria, or with a line containing infinitely many equilibrium points [1–18]. The attractors for such systems have been called “hidden attractors” [19–30], and that accounts for the difficulty of discovering them since there is no systematic way to choose initial conditions except by extensive numerical search. Hidden attractors are important in engineering applications because they allow unexpected and potentially disastrous responses to perturbations in a structure like a bridge or aircraft wing.

In this work we propose a new method for predicting the global behavior of chaotic flows with hidden attractors. It is known that the long-term prediction of chaotic time series is not possible due to the sensitive dependence on initial conditions [31] and that their prediction is much more difficult than for static/algebraic systems [32]. However it is still useful to find a model which can provide short-term prediction or can reproduce the geometrical properties of a chaotic system, such as the shape of the strange attractor.

Different approaches have been used for chaotic signal prediction. Fuzzy Function (FF) systems represent one of the recent interesting soft computing approaches used in various applications such as modeling, classification, and prediction [33]. Turksen introduced this type of fuzzy structure [33–35] which is simpler compared to neuro-fuzzy rule-based systems. The multidimensional input space of FFs leads to an elimination problem due to the projection onto each axis. This is one of the main differences between multidimensional structures and rule-based structures [36]. Consequently, the obtained membership values besides the input variables are used to estimate fuzzy functions. Different regression methods like Least Square Estimation (LSE) [35], Multi Adaptive Regression Spline (MARS) [37], and Support Vector Machine (SVM) [38] can be used to estimate these functions.

With the addition of recurrent structures to a model responding to memory information based on prior system states, a significant increase in addressing the temporal sequence capability can be achieved [39–44]. In this way, some literature exists on the combi-

* Corresponding author.
E-mail address: sajadjafari@aut.ac.ir (S. Jafari).

http://dx.doi.org/10.1016/j.physleta.2015.11.036
0375-9601/

nation of recurrent structures and fuzzy systems in two categories of local and global feedback. Juang et al. [41] employed local-rule feedback and took advantage of a variable-dimension Kalman filter for learning. Lin et al. [40] introduced recurrent self-evolving neuro-fuzzy networks that have a local and global link in the aggregation step. Ganjefar and Toghi placed a single neuron with a mother wavelet activation function and local feedback in each rule to achieve better results by modifying the learning algorithm especially in on-line applications [42]. Tellez et al. based on passivity theory and by the use of an online recurrent layer, introduced inverse optimal controllers which are trained by an extended Kalman filter [45]. Theocharis implemented recurrent structure with a context node in the form of an FIR synaptic filter and achieved enhanced temporal capacity in a higher-order system for modeling a complex nonlinear temporal process [39]. This paper proposes Recurrent Fuzzy Functions (RFFs) which have the following salient characteristics:

(1) A novel FF structure that benefits both interactive rules and recurrent structures is proposed. Before this work, FF systems were used with weighted averages based on rule firing rates for aggregation, but in this study interactively recurrent nodes are used to improve the learning capacity of the dynamical structures of a time series.

(2) One important task in designing recurrent systems is training of the feedback weights. In our proposed system nodes, weights are trained with steepest descent that automatically tune the learning rate with a line search based on the strong Wolf condition because of its fast learning speed.

(3) The computation is more efficient, and structure is simpler than other considered fuzzy structures, and it is more generalizable.

Also by use of RFFs as the nonlinear autoregressive with exogenous input (NARX) model of the data, prediction of chaotic flows with a hidden attractor is investigated using two different strategies: short-term (quantitative) and long-term (qualitative).

The rest of this paper is organized as follows: the preliminaries of the NARX model are briefly reviewed in section 2. Then in section 3, the FCM method for clustering will be presented. MARS regression is described in section 4. Details of the proposed RFFs' structure and parameter learning scheme are presented in section 5. In section 6, we introduce some rare chaotic systems with a hidden attractor which will be examined in case studies. Sections 7 and 8 give results and conclusions.

2. NARX model and optimal parameter

Generally in statistical prediction, a stochastic model based on previous observation is constructed to predict current and future values. A popular type of such a model is the nonlinear autoregressive moving average model with exogenous inputs (NARMAX) which is given by [46]:

$$y(t) = F\left[\, y(t-1), \ldots, y(t-n_y), e(t-1), \ldots, e(t-n_e), x(t-1), \ldots, x(t-n_x)\right] + e(t) \tag{1}$$

where x, e and y are external input, noise (which can be seen as representing the prediction error), and output of the system, respectively. F is an unknown nonlinear function, and n_x, n_e and n_y are the maximum lags of the input, noise, and output, respectively. A special case of the general model is the NARX (n_y, n_x) model:

$$y(t) = F\left[\, y(t-1), \ldots, y(t-n_y), x(t-1), \ldots, x(t-n_x)\right] + e(t) \tag{2}$$

where it is assumed that $e(t)$ has zero mean and finite variance σ^2 and is independent and identically distributed. The predictor model in many problems can be designed without the use of external input. Here we just use the past time series values and the prediction error.

The method of making a NARX representation involves determining the structure and estimating the parameters of the unknown nonlinear system from data. Here we use the proposed RFFs as the structure, and parameters are estimated that minimize the prediction error.

3. Fuzzy C-means clustering

The cost function of the basic FCM algorithm assuming a known number of clusters is as follows [47]:

$$J_q(V, U) = \sum_{i=1}^{M}\sum_{j=1}^{C} u_{ij}^{q} d(x_i, v_j) \tag{3}$$

subject to the constraints:

$$\sum_{j=1}^{C} u_{ij} = 1, \quad i = 1, \ldots, M \tag{4}$$

where

$$u_{ij} \in [0,1], \qquad 0 < \sum_{i=1}^{M} u_{ij} < M \quad i = 1, \ldots, M,\ j = 1, \ldots, C$$

and $q > 1$ is the fuzziness, and x_i, v_j, M, and C are the ith input data, the center of the jth cluster, the number of data points, and the number of clusters, respectively. Also U is an $M \times m$ matrix whose ijth element is the membership degree of x_i in the jth cluster, and V is a $C \times m$ matrix which contains the m-dimensional centers of the clusters, and $d(x_i, v_j)$ is the distance between x_i and the jth cluster center.

After minimization, a closed form for the degree of membership of the features in the clusters is as follows [47]:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{d(x_i, v_j)}{d(x_i, v_k)}\right)^{\frac{1}{q-1}}} \tag{5}$$

and the cluster prototype is:

$$v_j = \frac{\sum_{i=1}^{M} u_{ij}^{q} x_i}{\sum_{i=1}^{M} u_{ij}^{q}} \tag{6}$$

Fuzzy partition is carried out through an iterative process consisting of computing the degree of membership and center of the clusters by use of Eqs. (5) and (6), respectively, with random initialization. Based on the global convergence theorem of Zangwill [48], when different distance measures that satisfy certain conditions discussed in [49] are employed, convergence of the sequence produced by the above algorithm in a finite number of iterations to a local minima, has been proved [50].

4. Multi adaptive regression spline

MARS is one of the adaptive regression methods. This nonparametric regression approach can be considered a generalization of stepwise linear regression and can efficiently represent the nonlinear relation and hidden patterns in data sets [51]. The sum of squares error for a general regression is as follows:

$$SSE = \sum_{i=1}^{n}\left(y_i - \beta_0 - \beta_1 b_1(x_i) - \cdots - \beta_P b_P(x_i)\right)^2 \tag{7}$$

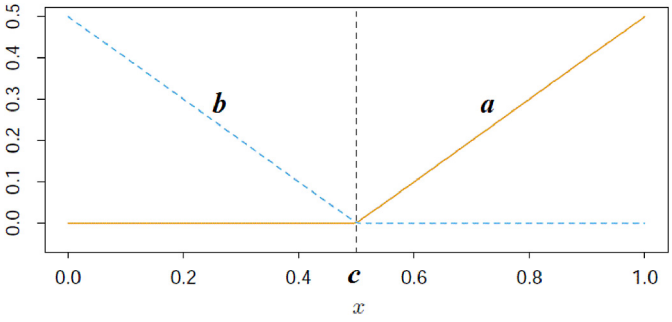

Fig. 1. Plot of spline functions [51] (a) h_1, (b) h_2.

where y is the dependent variable and $x = (x_1, \ldots, x_n)$ indicates the vector of explanatory variables in the observations where $x_i = (x_{i1}, x_{i2}, \ldots, x_{im}) \in R^m$ and $b_j(x)$ are basis functions. In this method, basis functions are formed by a pair of functions known as splines (h_1, h_2), given by [52]:

$$h_1 = \max\{0, x - c\}, \qquad h_2 = \max\{0, c - x\} \tag{8}$$

where c is a constant called the knot. Fig. 1 shows the spline functions.

If we have n independent observations for each of the explanatory variables x_j, each observation can be considered as a knot, and subsequently $2n$ functions exist for each x_j. Set B is defined so that it includes all functions h_{1j}, h_{2j}, $j = 1, \ldots, n$ and the product of two or more of them. For the basis function of Eq. (8), each member of set B can be used [52]. MARS is run in the forward and backward phase as follows:

Forward phase: This step begins with the mean of the dependent variable as b_0. Then an iterative process adds basis functions to the model. Estimation of the coefficients β is performed in such a way as to give the greatest decrease in the SSE. For adding a new statement, the algorithm must check all of the following items:

a) Basis functions exist so far in the model.
b) Variables exist so far in the model (can select one for making a basis function).
c) Variable values exist so far in the model (for specifying the knot in a new h function).

Adding new statements ends once changes in SSE become negligible or when a maximum number of statements is reached. At the end of this phase, a model with many statements is produced that has poor performance when generalized, and thus it needs a backward elimination phase.

Backward phase: Begin with the final model achieved in the previous phase, and at each iteration, remove the statement from the model that has the lowest contribution to an increased SSE. By pruning the model at each step, the result can be a candidate for the final model. "Generalized Cross Validation" is used for performance evaluation, and this procedure continues until all the basis functions are removed. MARS selects an optimum model which has a minimum GCV. The GCV for the ith step is as follows [52]:

$$GCV = \frac{RSS}{N \times \left(1 - \frac{\text{effective number of parameters}}{N}\right)^2} \tag{9}$$

$$\text{effective number of parameters} = \text{number of MARS terms} + \lambda \frac{\text{number of MARS terms} - 1}{2} \tag{10}$$

where $2 < \lambda < 4$ is the penalty value and N is the number of observations. Also $\frac{\text{number of MARS terms}-1}{2}$ is the number of knots of spline functions in the model.

5. Proposed method

The proposed RFFs is a modified version of FFs as shown in Fig. 2. The junctions between different rules give the system an interactively recurrent dynamical structure.

5.1. The structure formation

Suppose $X = [x_1 \ \cdots \ x_N]$ is the one-dimensional input time series. In state space reconstruction, with dimension m and lag τ, the trajectory matrix is obtained as follows:

$$S = \begin{bmatrix} x_1 & \cdots & x_{1+(m-1)\tau} \\ \vdots & \ddots & \vdots \\ x_{N-(m-1)\tau} & \cdots & x_N \end{bmatrix} \tag{11}$$

where N is the length of input time series. Each row of S is one point of the trajectory. So in the prediction task, the nth input data is $s(n) = (x(n), x(n+\tau), \ldots, x(n+(m-1)\tau))$, and the corresponding object is $y(n) = x(n+m\tau)$. RFFs model include the following steps:

Step 1: Trajectory points are clustered for determining the fuzzy rules (behaviors of the system). After termination of clustering, membership degrees of each state in each cluster is determined by Eq. (5).

Step 2: In this step, input data for regression in the next step are prepared. To this end, membership values and their transformations are appended to the primary inputs:

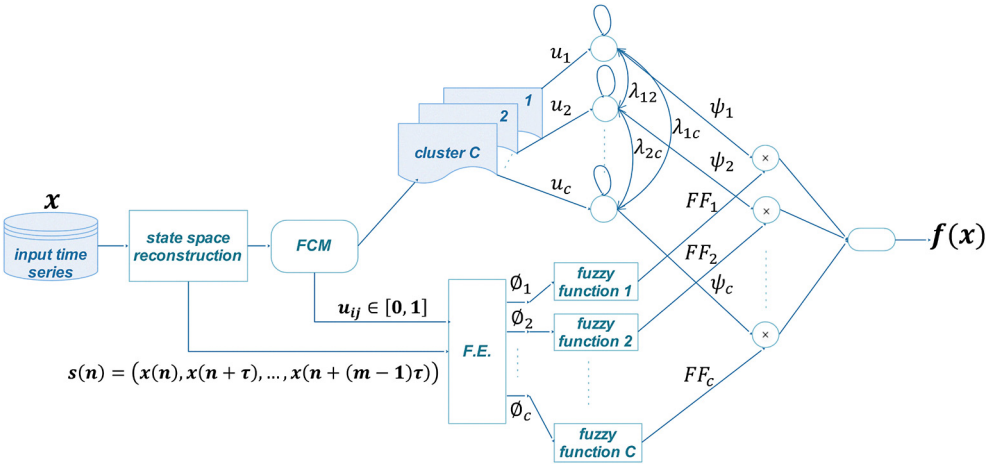

Fig. 2. The proposed RFFs framework.

$$\varphi_i = \left[s, \Gamma, f(\Gamma)\right] \quad i = 1, \ldots, M$$

where $M = N - (m-1)\tau$ is the number of points in state space and $f(\Gamma)$ indicates the membership value transformation (more detail about this step is in [35]).

Step 3: MARS is applied to matrix ϕ in order to determine the Fuzzy Functions as follows:

$$\phi_i = \begin{bmatrix} s_1 & \Gamma_1 & f(\Gamma_1) \\ \vdots & \vdots & \vdots \\ s_K & \Gamma_K & f(\Gamma_K) \end{bmatrix} \quad i = 1, \ldots, C \tag{12}$$

where K is the number of the cluster's point and for each row (explanatory variable) and its corresponding output is $x_{n+m\tau}$. Here, a NARX model for all clusters has been made which corresponds to the behavior of each cluster.

Step 4: Recurrent layer of RFFs are considered to form local and global feedback loops. The obtained weights of this step are used for aggregation between the different rules. These weights lead to a temporal memory and are computed as follows:

$$\psi_k^t(u) = (1-\gamma_k)u_k^t + \sum_{\substack{i=1 \\ i \neq k}}^{C} \lambda_{ik}\psi_i^{t-1}$$
$$k = 1, \cdots, C, \; t = 1, \cdots, M \tag{13}$$

Here $\gamma_k = \sum_{i=1}^{C} \lambda_{ik}$, u_k^t, t, ψ, and λ are the membership values of the tth data in cluster k, the number of samples, the weight of aggregation, and the coefficient of interactive and recurrent relations, respectively.

Step 6: Finally, by weighted averaging, the system output is calculated as follows:

$$f(x_j) = \frac{\sum_{i=1}^{C} \psi_i(u) . FF_i(\varphi_j)}{\sum_{i=1}^{C} \psi_i(u)} \tag{14}$$

where $FF_i(\varphi_j)$ is the output of the ith rule for the jth input point.

5.2. Parameter learning

Here parameter learning of the proposed system is described. Parameters are recurrent layer coefficients matrix (λ) as follows:

$$\lambda = \begin{bmatrix} \lambda_{11} & \cdots & \lambda_{1C} \\ \vdots & \ddots & \vdots \\ \lambda_{C1} & \cdots & \lambda_{CC} \end{bmatrix} \tag{15}$$

Here we have an essential assumption that the interactive weight is bidirectional, so λ is symmetric and elements on the main diagonal indicate recurrent weights, and other elements indicate interactive weights. As mentioned in section 1, based on the error of the modeling parameter, learning is done in an iterative algorithm. The prediction error value is as follows:

$$E = \frac{1}{2}\left(y^t - f\left(x^t\right)\right)^2 \tag{16}$$

where $f(x^t)$ is the RFFs output value for the tth sample and y^t is the desired output value. Coefficients train using the gradient descent algorithm with a line search based on the strong Wolfe condition, and its main advantage is automatic determination of the learning rate. In the basic gradient descent algorithm without a line search, large η leads to divergence and to increased learning time and computation. In addition, accuracy of the result is never greater than η. If the ideal line search method is used, it might have further computation costs because a univariate optimization method (like the Golden Section Search) must be used to find the optimal value of η. That is an iterative process like the original optimization method, and so it requires more computation. But if a line search based on the strong Wolfe condition used, it could determine an optimal value of η in a few iterations [53]. Results in the next section demonstrate this superiority. Coefficients update in the following form:

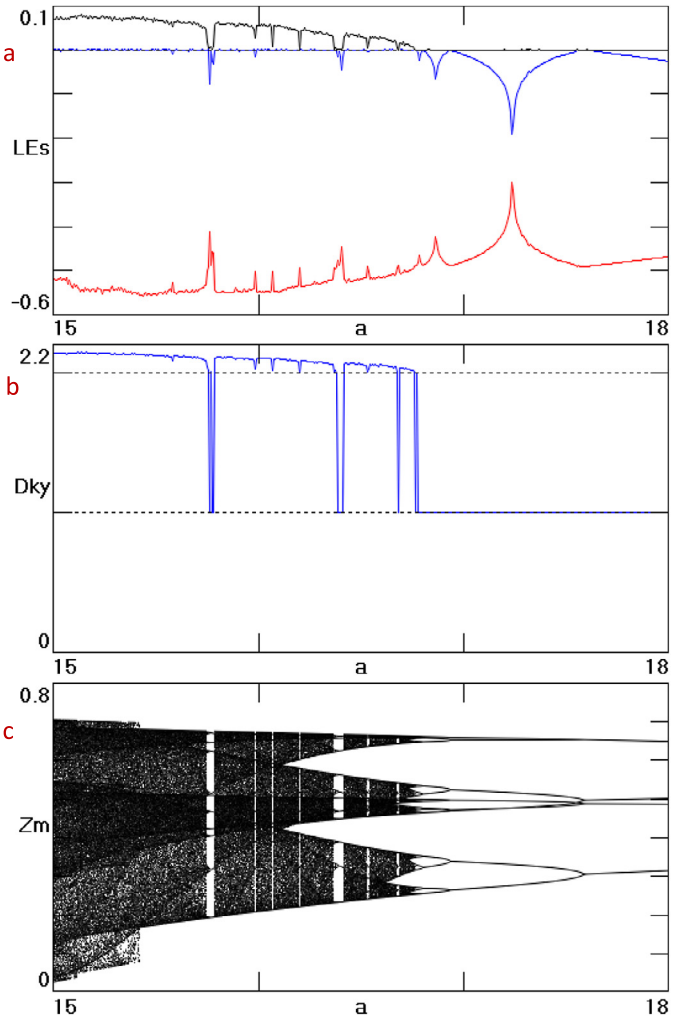

Fig. 3. (a) Lyapunov exponents, (b) Kaplan–Yorke dimension, and (c) bifurcation diagram of system (20) showing a period-doubling route to chaos.

$$\lambda_{ij}^{t+1} = \lambda_{ij}^t + \Delta\lambda_{ij}^t + \propto \Delta\lambda_{ij}^{t-1} \tag{17}$$

$$\Delta\lambda_{ij}^t = -\eta\frac{\partial E}{\partial\lambda_{ij}} \quad i, j = 1, \cdots, K \tag{18}$$

where $\propto$ is the momentum weight (which is practically selected around 0.75) and $0 < \eta < 1$ is the learning step. Developing the previous expressions gives:

$$\frac{\partial E}{\partial\lambda_{ij}} = -\left(y - f(x)\right)\frac{\partial f}{\partial\lambda_{ij}} \tag{19}$$

where

$$\frac{\partial f}{\partial\lambda_{ij}} = \begin{cases} \left(\psi_i(t-1) - u_i(t)\right) FF_i(q) & i = j \\ \left(\psi_j(t-1) - u_i(t)\right) FF_i(q) \\ \quad + \left(\psi_i(t-1) - u_j(t)\right) FF_j(q) & i \neq j \end{cases} \tag{20}$$

6. Rare chaotic flows with hidden attractors

In this section, we introduce some new rare three-dimensional chaotic systems which are proposed by Jafari and Sprott [1], Jafari

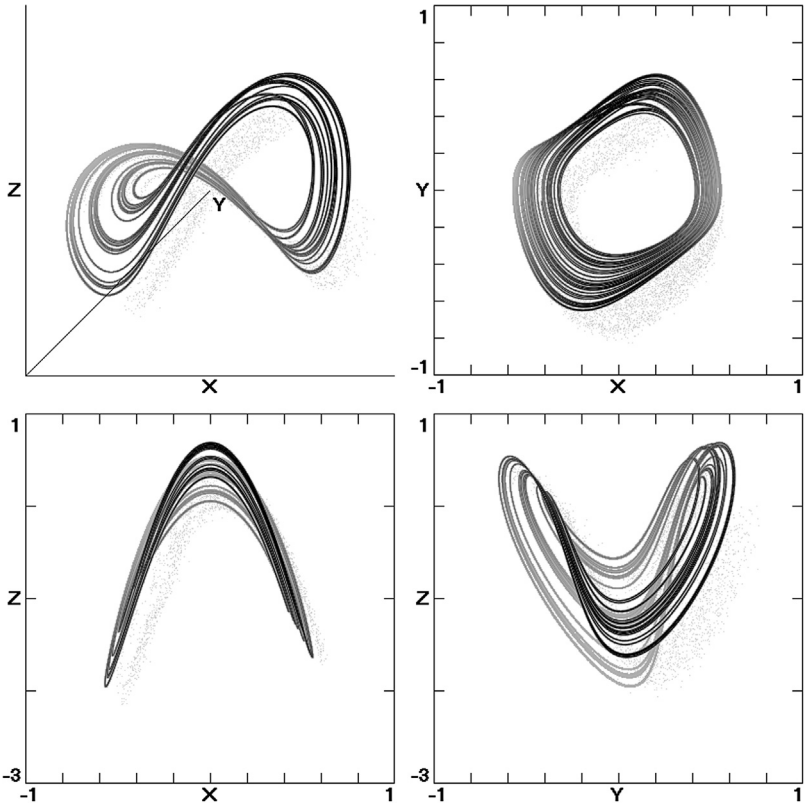

Fig. 4. Four views of the strange attractor in Eq. (22) with initial conditions $(0, 0.5, 0.5)$.

et al. [3], Sprott et al. [14], and Molaie et al. [7]. First we investigate in detail the simplest chaotic system with a line of equilibria, and then we mention three other chaotic flows with hidden attractors. Consider the following system [1]:

$$\begin{aligned}\dot{x} &= y\\ \dot{y} &= -x + yz\\ \dot{z} &= -x - axy - bxz\end{aligned} \tag{21}$$

Choosing the most elegant set of parameters [54] gives the following system:

$$\text{Sys 1:}\quad \begin{aligned}\dot{x} &= y\\ \dot{y} &= -x + yz\\ \dot{z} &= -x - 15xy - xz\end{aligned} \tag{22}$$

The equilibrium states of this system are $E = (0, 0, z)$, which actually constitutes a line. The eigenvalues of these equilibria are $(0, (z \pm \sqrt{z^2 - 4})/2)$. The dynamical behavior of system (20) can be illustrated by numerical bifurcation diagrams, Lyapunov exponents, phase portraits and time series. Fig. 3 shows the Lyapunov exponents, Kaplan–Yorke dimension, and the bifurcation diagram depicting maxima of $z(t)$ versus the parameter a for $b = 1$.

The chaotic behavior shown in Fig. 3 is further detailed in Fig. 4 which shows the trajectory projected onto various planes using initial conditions $(0, 0.5, 0.5)$ for $a = 15$ and $b = 1$. Lyapunov exponents are $(0.0717, 0, -0.5232)$, and the Kaplan–Yorke dimension is $D_{KY} = 2.1371$. Cross sections of the basin of attraction for the system are shown in Fig. 5.

In addition, we consider three other rare systems with hidden attractors. System (2) [3] is a system with no equilibria, System (3) [7] is a system with one stable equilibrium, and System (4) [14] is a system with one unstable node:

$$\text{Sys 2:}\quad \begin{aligned}\dot{x} &= -y\\ \dot{y} &= x + z\\ \dot{z} &= 2y^2 + xz - 0.35\end{aligned} \tag{23}$$

$$\text{Sys 3:}\quad \begin{aligned}\dot{x} &= y\\ \dot{y} &= z\\ \dot{z} &= -x - 0.6y - 2z + z^2 - 0.4xy\end{aligned} \tag{24}$$

$$\text{Sys 4:}\quad \begin{aligned}\dot{x} &= y\\ \dot{y} &= -x + yz\\ \dot{z} &= z + 8.888x^2 - y^2 - 4\end{aligned} \tag{25}$$

7. Simulation results

The problem of time series prediction is considered as a modeling problem containing three steps. First, past and present values of the time series are considered as inputs, and future values are considered as outputs of the prediction model. Next a model relating output to input is built. Finally, the created model is used to predict unseen future points [55]. As mentioned before, the goal of this paper is to examine the performance of the recurrent fuzzy functions structure for a chaotic time series with a hidden attractor in long term and short term prediction. In addition to the four new chaotic flows introduced in the previous section, we examine the method for two classical types of chaotic flows, the first of which is the Lorenz system given by [56]:

$$\text{Sys 5:}\quad \begin{aligned}\dot{x} &= a(y - x)\\ \dot{y} &= cx - xz - y\\ \dot{z} &= xy - bz\end{aligned} \tag{26}$$

where x, y, and z are state variables and a, b, and c are adjustable parameters. For $a = 10$, $b = 8/3$, and $c = 28$ the Lorenz equations will give chaotic behavior.

Another classical system is the Rössler system given by [57]:

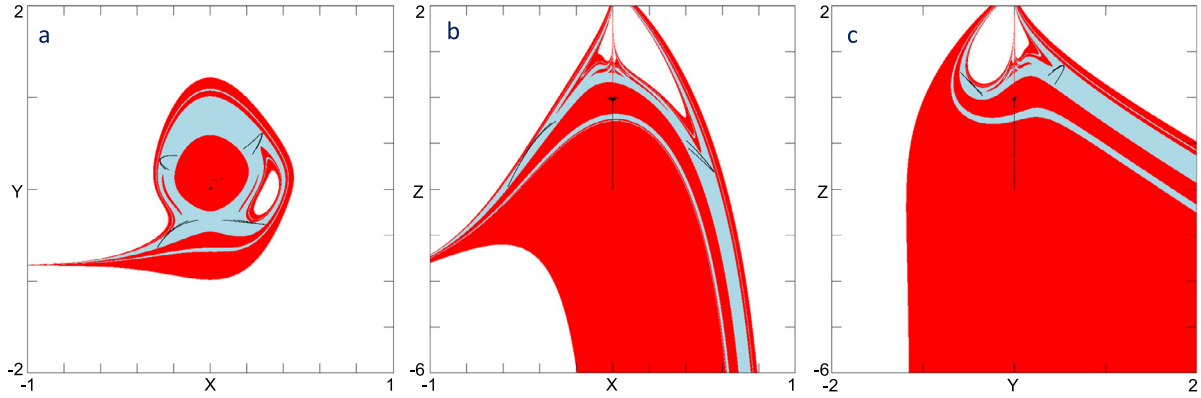

Fig. 5. Basin of attraction of system (22) (a) in the plane $z = 0$, (b) in the plane $y = 0$, and (c) in the plane $x = 0$. Initial conditions in the white region lead to unbounded orbits, those in the light blue region lead to the strange attractor, and those in the red region lead to the line equilibrium. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\text{Sys 6:}\quad \begin{aligned}\dot{x} &= -z - y\\ \dot{y} &= x + ay\\ \dot{z} &= b + zx - cz\end{aligned} \tag{27}$$

where typical values of $a = 0.15$, $b = 0.20$, and $c = 10$ give chaotic behavior.

Simulation results for the six systems are presented in two subsections depending on the prediction time. In addition, performance of the proposed method is compared with other methods, including fuzzy functions with least square error (FF-LSE) [35], fuzzy functions with multi-adaptive regression spline (FF-MARS) [37] and multi-adaptive regression spline (MARS) [52]. The percent root mean squared difference (PRD) is taken as the quality criteria:

$$PRD\% = \sqrt{\frac{\sum_{i=1}^{N}(y(i) - \hat{y}(i))^2}{\sum_{i=1}^{N}(y(i))^2}} \times 100\% \tag{28}$$

where $\hat{y}$ is the predicted output. Choosing optimal parameters and a time step of 0.05 and sampling from 0 to 100 second with appropriate initialization of the variables gives a 2000-point chaotic time series $x(t)$, $y(t)$, and $z(t)$ as seen in Fig. 6 for Sys 1 (Eq. (22)) which is discussed in detail. In the rest of this section, we used the x of each system with state space reconstruction (SSR). In the simulation we used the first 80 seconds (1600 points) as training data and the remaining 400 points for testing (in the interest of brevity, we only show results for Sys 1).

7.1. One step ahead (short-term) prediction

Two time series prediction problems are: short-term (one step ahead) and long term (many steps ahead). In the first type, we test the proposed method using two different approaches:

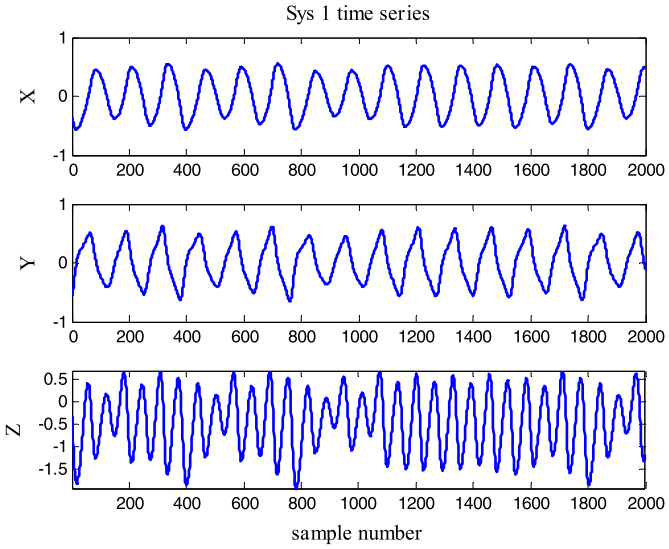

Fig. 6. Time series of system (22).

7.1.1. Without state space reconstruction

In this subsection we predict future results with a delay equal to 1. Here we make a 3rd order NARX model. Table 1 shows the results and a graphical performance comparison of the different methods for all the systems provided in Fig. 7.

Table 1 and Fig. 7 show that simpler methods like FF-LSE, although having weaker results in the training part, do not have a large reduction in the test part unlike the FF-MARS. Table 1 also indicates that by the addition of a recurrent layer and interactive structure in the RFFs, better PRDs than those of other fuzzy structures are achieved. Although in RFFs, the number of parameters increased, despite expectations, generalization of the system in the

Table 1
Results of short-term prediction without SSR.

Performance		Method							
		FF-LSE		FF-MARS		MARS		**RFFs**	
		Train	Test	Train	Test	Train	Test	**Train**	**Test**
PRD	Sys 1	0.996	0.023	0.028	0.030	2.544	2.782	**0.021**	**0.017**
	Sys 2	0.406	0.018	0.064	0.07	1.573	1.539	**0.058**	**0.064**
	Sys 3	0.256	0.027	0.585	0.834	1.936	1.864	**0.584**	**0.832**
	Sys 4	1.048	0.631	0.719	0.655	1.689	1.738	**0.710**	**0.645**
	Sys 5	0.59	0.324	0.961	0.774	1.118	1.943	**0.73**	**0.376**
	Sys 6	0.328	0.176	0.846	0.362	1.004	1.871	**0.409**	**0.206**

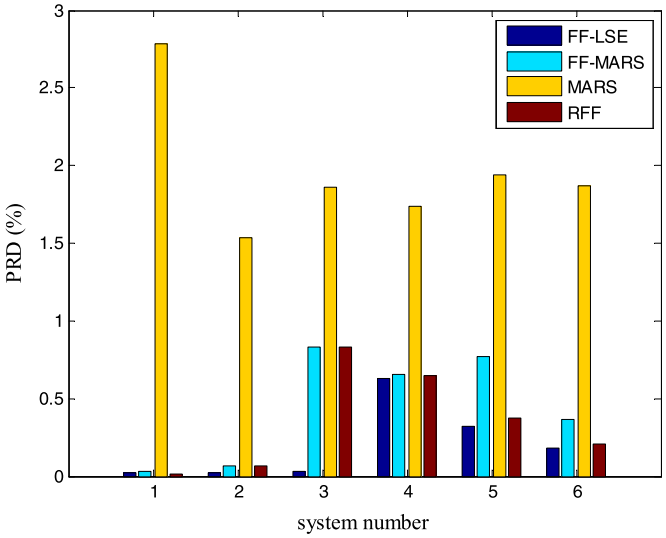

Fig. 7. Comparison of different methods for *all the systems in short-term prediction without SSR.*

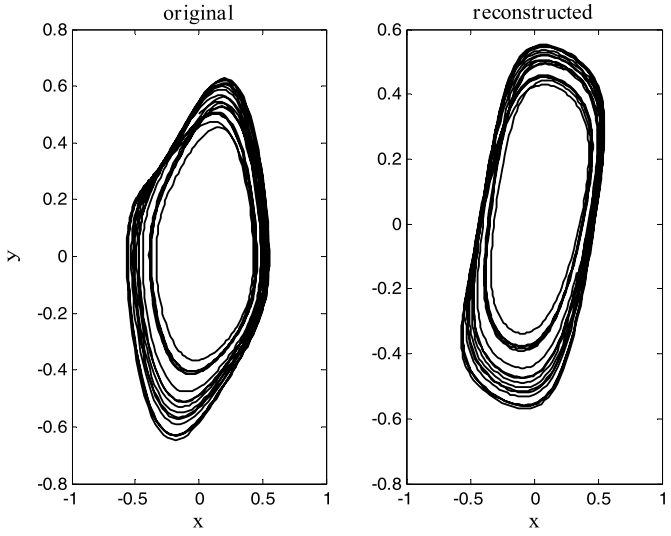

Fig. 8. State space plots projected onto the xy-plane.

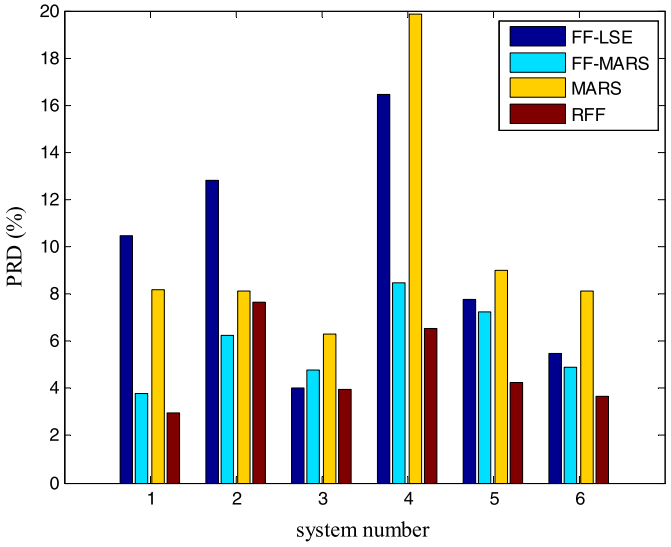

Fig. 9. Performance comparison for *all the systems in short term prediction with SSR.*

test part do not decrease. To corroborate this point, it should be said that interactive and recurrent weights learn the time pattern of a sequence (memory information), and so it is effective at obtaining better test results.

7.1.2. With state space reconstruction

In most practical processes, all of the dynamical/state variables are not accessible, and only a single time series of observation is available. In such a case, additional variables can be generated using state space reconstruction as introduced in the literature many times [58,59], and thus will not be repeated here. However, it is necessary to choose an embedding dimension and a time step. The embedding dimension is determined using the method of false nearest neighbors (FNN) [60] from which we conclude that a value of 3 is sufficient (for all systems). For determining an optimal time step, we used the first local minimum of the mutual information function [61] which gives a value of 24 (for system 1). State space plots of the system projected onto the xy-plane in the original case and achieved through state space reconstruction are shown in Fig. 8.

Fig. 8 confirms the Takens theory [59] and shows the homology of the original and reconstructed attractor. Table 2 shows the results achieved for prediction based on SSR. Fig. 9 which contains the bar plot of Table 2, shows that when the behavior of the system is recovered by SSR, the MARS method has better performance than the FF-LSE method because of the greater power of the MARS method compared to LSE in regression. Combing the two things: a) fuzzy ability in extracting different behaviors in a dynamical systems and b) MARS ability in function approximation, provides a powerful tool for our goal. Finally, by adding recurrent structure (as memory) to this system, we get the best result.

As seen in Tables 1 and 2, future points predicted without SSR give better results (lower error), but its generalization to long-term prediction fails. In fact, an advantage of SSR is that a time series with random features can appear deterministic in a reconstructed state space [62]. Based on this point, we use SSR for long-term prediction.

Table 2
Results of short-term prediction with SSR.

Performance		Method							
		FF-LSE		FF-MARS		MARS		**RFFs**	
		Train	Test	Train	Test	Train	Test	**Train**	**Test**
PRD	Sys 1	9.8838	10.462	3.7398	3.7913	7.7630	8.1901	**2.8227**	**2.9568**
	Sys 2	14.3531	12.812	8.955	6.27	10.402	8.147	**7.393**	**7.658**
	Sys 3	2.984	4.044	3.521	4.771	4.747	6.276	**3.127**	**3.984**
	Sys 4	15.459	16.429	8.209	8.503	18.587	19.861	**6.394**	**6.567**
	Sys 5	6.859	7.793	6.993	7.245	8.815	9.005	**3.962**	**4.259**
	Sys 6	4.925	5.471	3.195	4.89	7.391	8.11	**3.18**	**3.667**

Table 3
Results of long-term prediction using recursion.

Performance		Method			
		FF-LSE	FF-MARS	MARS	**RFFs**
		Test	Test	Test	**Test**
PRD	Sys 1	14.7806	19.7680	11.7276	**7.8321**
	Sys 2	21.432	33.1435	20.009	**16.4192**
	Sys 3	16.1917	24.661	16.321	**10.7359**
	Sys 4	20.4325	29.19	17.012	**15.0184**
	Sys 5	21.1462	23.0145	14.2695	**9.0241**
	Sys 6	18.3461	22.486	13.674	**8.5364**

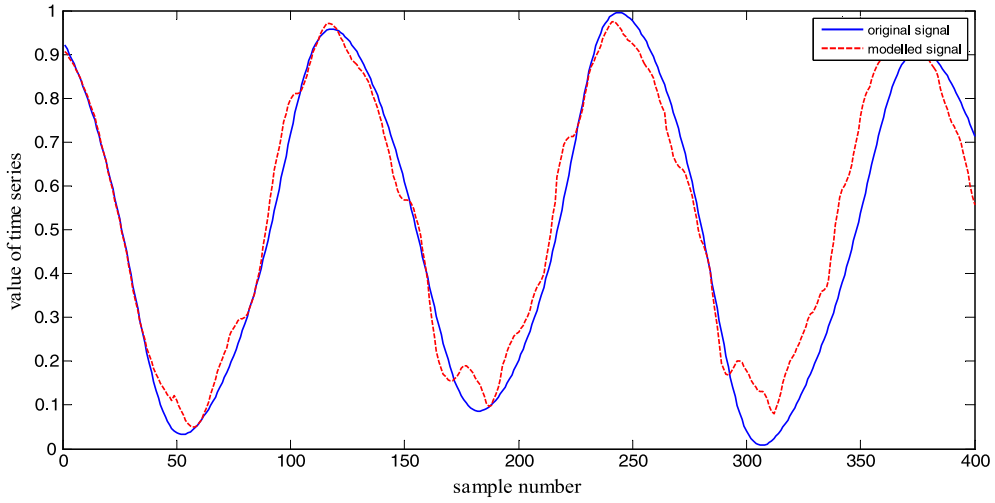

Fig. 10. Long-term prediction results for the X time series using the recursive approach (system 1).

7.2. Multi-step (long-term) prediction

The problem of long-term prediction is challenging. Results of long-term prediction are inaccurate and unreliable due to uncertainties and error propagation. Long-term forecasting methods usually extend short-term prediction models by two different strategies: recursive prediction [63] and direct prediction (multi-model) [64].

7.2.1. Recursive prediction

Here after making a model in the training part, we use it recursively in the test section. In this way, to predict several steps ahead we use the predicted values as known data to predict the next value in the sample. In more detail, the procedure can be constructed by first making one-step ahead prediction based on SSR [55]:

$$\hat{x}(n+m\tau) = f\big(x(n), x(n+\tau), \ldots, x\big(n+(m-1)\tau\big)\big) \tag{29}$$

where m denotes the dimension. To predict the next value, the approximator of Eq. (14) is defined as the vector of inputs which contain $\hat{x}(n+m\tau)$ instead of the true values, and similarly for all future points. Fig. 11 compares achieved results by the different methods. Table 3 shows achieved results for 400 points ahead in the recursive long-term approach. As seen in Fig. 10, the error gradually increases in time due to error propagation, and this continues until complete divergence of the modeled signal occurs [63]. In this way, based on initial error of the model and the degree of trajectory divergence (Largest Lyapunov Exponent), the maximum prediction horizon can be calculated [65].

7.2.2. Multiple model approach

In contrast to the last subsection which used just one model for long-term prediction, here inspired by [66], we train 50 models for

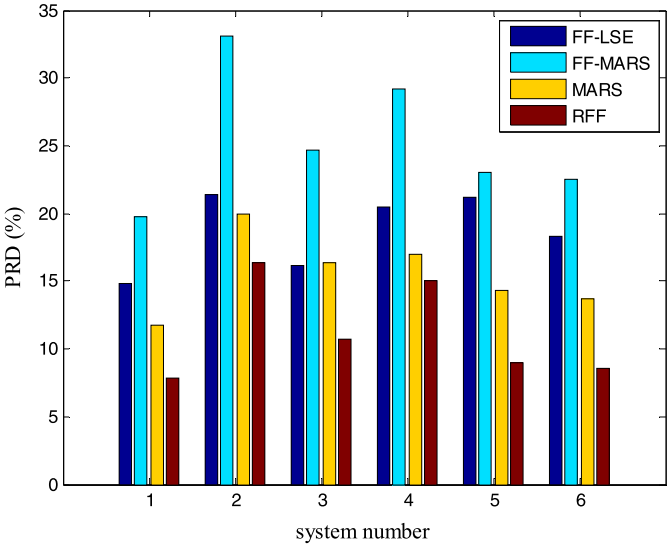

Fig. 11. Comparison of different methods for *all the systems using the recursive approach.*

long-term prediction. In this way, each model trains for predicting independent points ahead, while having the same inputs as bellow:

$$\hat{x}(n+P) = f_P\big(x(n), x(n+\tau), \ldots, x\big(n+(m-1)\tau\big)\big) \tag{30}$$

Using a number of models equal to the prediction horizon, we used all models together for prediction points after the prediction horizon in the recursive mode. Using a further model, we prevent divergence and accumulation of error. As can be seen in Fig. 12, in contrast to Fig. 10, the prediction error for all points remains at a certain level, and by increasing the number of models,

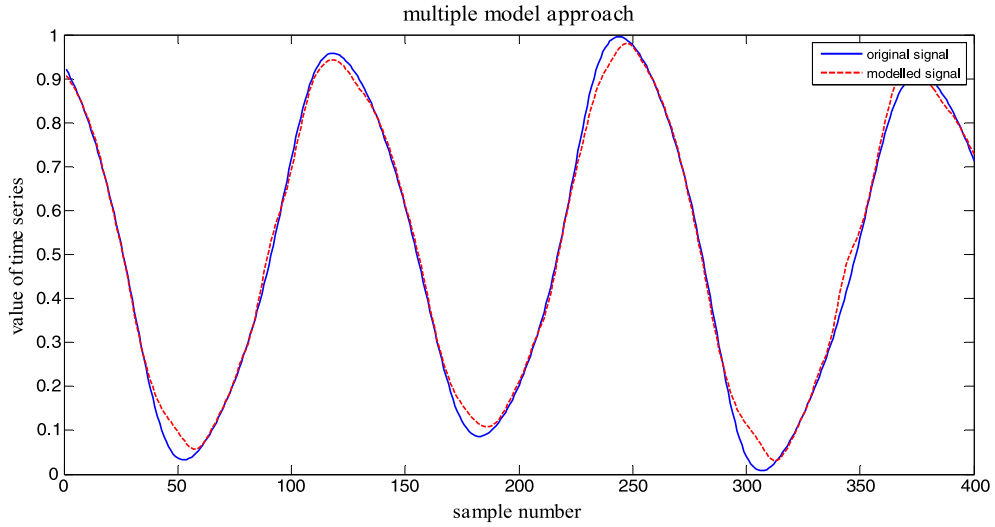

Fig. 12. Long-term prediction result for X time series by multiple model approach (system 1).

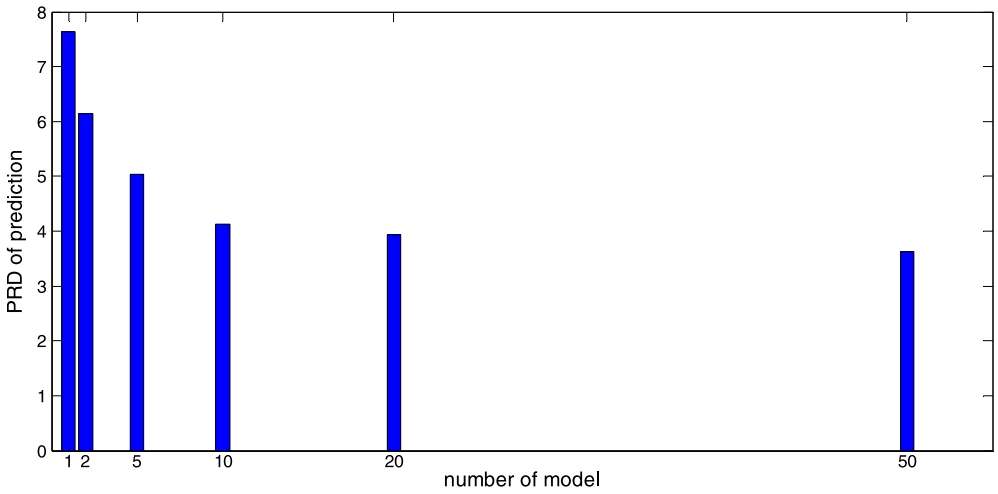

Fig. 13. Long-term prediction error reduction based on increasing the number of models in multiple model approach.

Table 4
Results of long-term prediction using multiple model prediction.

Performance		Method			
		FF-LSE	FF-MARS	MARS	**RFFs**
		Test	Test	Test	**Test**
PRD	Sys 1	16.006	10.4123	8.106	**4.7854**
	Sys 2	29.132	26.864	14.367	**11.961**
	Sys 3	21.741	15.356	11.005	**6.846**
	Sys 4	24.132	19.174	13.112	**9.058**
	Sys 5	21.639	16.376	10.837	**7.527**
	Sys 6	19.485	13.81	9.0142	**5.3192**

the prediction performance can be improved. Fig. 13 shows the obtained PRD measure as a function of the number of models. Based on Eq. (30), as P increases, the distance between input samples and the object of the model increases and this leads to an increase in error, and for this reason, the error is not reduced from a certain amount, and also it can be said that unlimited increase in the number of models could not lead to zero error, and however we have a certain degree of error because of further parameters. Table 4 shows achieved performance measure, and Fig. 14 compares achieved results by the different methods using 50 models.

As seen in Tables 3 and 4, the proposed RFFs method has the best results compared to other approaches for long-term prediction. As can be seen, the results of prediction using FF-MARS and RFF are similar for short-term prediction, but RFF has the best results in long-term prediction. To explain this achieved result, it can be said that by adding recurrent layer, the model extracts memory information and so has a powerful capability of learning the dynamical behavior of the system. This leads to a better tracking and prevents divergence in the long term.

One of the important issues that must be investigated is computational cost of the different methods. Based on our analysis, different methods have various learning times in the training part, but no appreciable difference exists in the test part. As seen in Fig. 15, the best method is FF-LSE in which fuzzy functions are

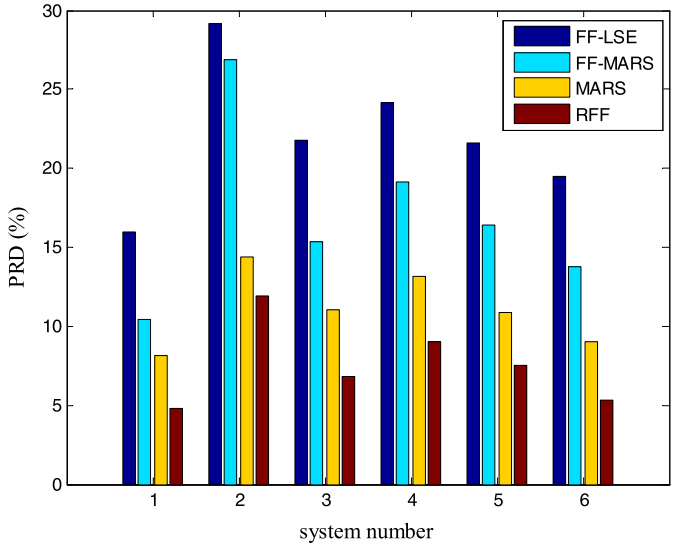

Fig. 14. Comparison of different methods for *all the systems in the multiple model approach*.

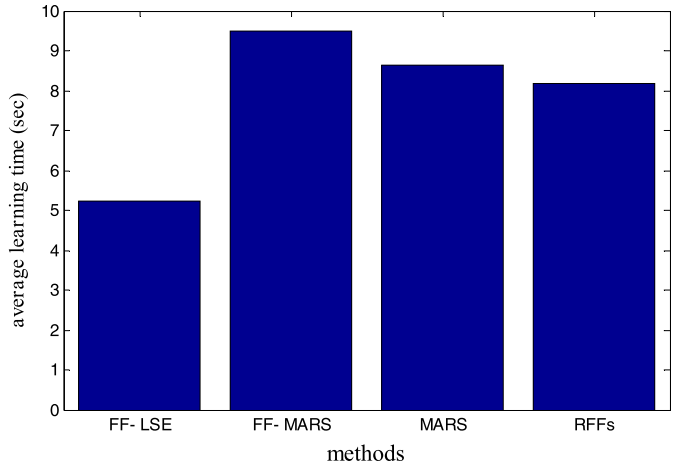

Fig. 15. Average computational cost of different methods for all prediction scenarios and systems.

trained in a closed form matrix computation. However, as can be seen in the results, this low computational cost leads to lower performance. The other three methods do not have appreciable differences.

8. Conclusion

In this paper, a global approach called Recurrent Fuzzy Functions (RFFs) for nonlinear modeling of a time series is proposed. Its performance is tested and compared on some new chaotic flows, especially newly introduced chaotic systems with hidden attractors. The proposed RFF works as a nonlinear autoregressive with exogenous inputs (NARX) model. It provides a kind of step by step model generation. Finally, by tuning the interactive weights, the relation between each subspace (clusters) of state space is determined. On the other hand, by tuning recurrent weights, the model can learn the dynamics of the original system (which is reflected in the observed time series). Experiments and comparative studies demonstrate better performance of the proposed approach over some other existing methods, in both short-term and long-term prediction. The simulations indicate that multiple models show a better prediction accuracy in the long term. However even when the accuracy fails in the time domain (due to the butterfly effect), the shape of the strange attractor obtained from the model matches the original strange attractor of the system. This can help in extracting some qualitative information related to the geometry of the attractors.

References

[1] S. Jafari, J. Sprott, Simple chaotic flows with a line equilibrium, Chaos Solitons Fractals 57 (2013) 79–84.
[2] S. Jafari, J. Sprott, Simple chaotic flows with a line equilibrium, Chaos Solitons Fractals 57 (2013) 79–84; Erratum, Chaos Solitons Fractals 77 (2015) 341–342.
[3] S. Jafari, J. Sprott, S.M.R.H. Golpayegani, Elementary quadratic chaotic flows with no equilibria, Phys. Lett. A 377 (2013) 699–702.
[4] S. Jafari, J.C. Sprott, V.-T. Pham, S.M.R.H. Golpayegani, A.H. Jafari, A new cost function for parameter estimation of chaotic systems using return maps as fingerprints, Int. J. Bifurc. Chaos Appl. Sci. Eng. 24 (2014) 1450134.
[5] S. Kingni, S. Jafari, H. Simo, P. Woafo, Three-dimensional chaotic autonomous system with only one stable equilibrium: analysis, circuit design, parameter estimation, control, synchronization and its fractional-order form, Eur. Phys. J. Plus 129 (2014) 1–16.
[6] S.-K. Lao, Y. Shekofteh, S. Jafari, J.C. Sprott, Cost function based on gaussian mixture model for parameter estimation of a chaotic circuit with a hidden attractor, Int. J. Bifurc. Chaos Appl. Sci. Eng. 24 (2014) 1450010.
[7] M. Molaie, S. Jafari, J.C. Sprott, S.M.R.H. Golpayegani, Simple chaotic flows with one stable equilibrium, Int. J. Bifurc. Chaos Appl. Sci. Eng. 23 (2013) 1350188.
[8] V.-T. Pham, S. Jafari, C. Volos, X. Wang, S.M.R.H. Golpayegani, Is that really hidden? The presence of complex fixed-points in chaotic flows with no equilibria, Int. J. Bifurc. Chaos Appl. Sci. Eng. 24 (2014) 1450146.
[9] V.-T. Pham, S. Vaidyanathan, C. Volos, S. Jafari, Hidden attractors in a chaotic system with an exponential nonlinear term, Eur. Phys. J. Spec. Top. 224 (2015) 1507–1517.
[10] V.-T. Pham, C. Volos, S. Jafari, X. Wang, Generating a novel hyperchaotic system out of equilibrium, Optoelectron. Adv. Mater., Rapid Commun. 8 (2014) 535–539.
[11] V.-T. Pham, C. Volos, S. Jafari, X. Wang, S. Vaidyanathan, Hidden hyperchaotic attractor in a novel simple memristive neural network, Optoelectron. Adv. Mater., Rapid Commun. 8 (2014) 1157–1163.
[12] V.-T. Pham, C. Volos, S. Jafari, Z. Wei, X. Wang, Constructing a novel no-equilibrium chaotic system, Int. J. Bifurc. Chaos Appl. Sci. Eng. 24 (2014) 1450073.
[13] M. Shahzad, V.-T. Pham, M. Ahmad, S. Jafari, F. Hadaeghi, Synchronization and circuit design of a chaotic system with coexisting hidden attractors, Eur. Phys. J. Spec. Top. 224 (2015) 1637–1652.
[14] J.C. Sprott, S. Jafari, V.-T. Pham, Z.S. Hosseini, A chaotic system with a single unstable node, Phys. Lett. A 379 (9/25/2015) 2030–2036.
[15] F.R. Tahir, S. Jafari, V.-T. Pham, C. Volos, X. Wang, A novel no-equilibrium chaotic system with multiwing butterfly attractors, Int. J. Bifurc. Chaos Appl. Sci. Eng. 25 (2015) 1550056.
[16] X. Wang, G. Chen, A chaotic system with only one stable equilibrium, Commun. Nonlinear Sci. Numer. Simul. 17 (2012) 1264–1272.
[17] X. Wang, G. Chen, Constructing a chaotic system with any number of equilibria, Nonlinear Dyn. 71 (2013) 429–436.
[18] Z. Wei, Dynamical behaviors of a chaotic system with no equilibria, Phys. Lett. A 376 (2011) 102–108.
[19] V. Bragin, V. Vagaitsev, N. Kuznetsov, G. Leonov, Algorithms for finding hidden oscillations in nonlinear systems. The Aizerman and Kalman conjectures and Chua's circuits, J. Comput. Syst. Sci. Int. 50 (2011) 511–543.
[20] N. Kuznetsov, G. Leonov, Hidden attractors in dynamical systems: systems with no equilibria, multistability and coexisting attractors, in: IFAC World Congress, vol. 19, 2014, pp. 5445–5454.
[21] G. Leonov, N. Kuznetsov, Analytical-numerical methods for investigation of hidden oscillations in nonlinear control systems, in: IFAC Proceedings Volumes (IFAC-PapersOnline), vol. 18, 2011, pp. 2494–2505.
[22] G. Leonov, N. Kuznetsov, M. Kiseleva, E. Solovyeva, A. Zaretskiy, Hidden oscillations in mathematical model of drilling system actuated by induction motor with a wound rotor, Nonlinear Dyn. 77 (2014) 277–288.
[23] G. Leonov, N. Kuznetsov, T. Mokaev, Homoclinic orbit and hidden attractor in the Lorenz-like system describing the fluid convection motion in the rotating cavity, preprint, arXiv:1412.7667, 2014.
[24] G. Leonov, N. Kuznetsov, T. Mokaev, Hidden attractor and homoclinic orbit in Lorenz-like system describing convective fluid motion in rotating cavity, Commun. Nonlinear Sci. Numer. Simul. 28 (2015) 166–174.
[25] G. Leonov, N. Kuznetsov, T. Mokaev, Homoclinic orbits, and self-excited and hidden attractors in a Lorenz-like system describing convective fluid motion, preprint, arXiv:1505.04729, 2015.
[26] G. Leonov, N. Kuznetsov, V. Vagaitsev, Localization of hidden Chua's attractors, Phys. Lett. A 375 (2011) 2230–2233.
[27] G. Leonov, N. Kuznetsov, V. Vagaitsev, Hidden attractor in smooth Chua systems, Physica D 241 (2012) 1482–1486.

[28] G.A. Leonov, N.V. Kuznetsov, Hidden attractors in dynamical systems. From hidden oscillations in Hilbert–Kolmogorov, Aizerman, and Kalman problems to hidden chaotic attractor in Chua circuits, Int. J. Bifurc. Chaos Appl. Sci. Eng. 23 (2013).
[29] P. Sharma, M. Shrimali, A. Prasad, N. Kuznetsov, G. Leonov, Control of multistability in hidden attractors, Eur. Phys. J. Spec. Top. 224 (2015) 1485–1491.
[30] P.R. Sharma, M.D. Shrimali, A. Prasad, N. Kuznetsov, G. Leonov, Controlling dynamics of hidden attractors, Int. J. Bifurc. Chaos Appl. Sci. Eng. (2015).
[31] R.C. Hilborn, Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers, Oxford University Press, Oxford, 2000.
[32] H. Kantz, T. Schreiber, Nonlinear Time Series Analysis, vol. 7, Cambridge University Press, 2004.
[33] I.B. Turksen, A review of developments from fuzzy rule bases to fuzzy functions, Hacet. J. Math. Stat. 41 (2012).
[34] A. Celikyilmaz, I.B. Turksen, Enhanced fuzzy system models with improved fuzzy clustering algorithm, IEEE Trans. Fuzzy Syst. 16 (2008) 779–794.
[35] I.B. Türkşen, Fuzzy functions with LSE, Appl. Soft Comput. 8 (2008) 1178–1188.
[36] O. Uncu, L. Turksen, A novel fuzzy system modeling approach: multidimensional structure identification and inference, in: The 10th IEEE International Conference on Fuzzy Systems 2001, 2001, pp. 557–561.
[37] M. Zarandi, M. Zarinbal, N. Ghanbari, I. Turksen, A new fuzzy functions model tuned by hybridizing imperialist competitive algorithm and simulated annealing. Application: stock price prediction, Inf. Sci. 222 (2013) 213–228.
[38] A. Çelikyılmaz, I. Burhan Türkşen, Fuzzy functions with support vector machines, Inf. Sci. 177 (2007) 5163–5177.
[39] J. Theocharis, A high-order recurrent neuro-fuzzy system with internal dynamics: application to the adaptive noise cancellation, Fuzzy Sets Syst. 157 (2006) 471–500.
[40] Y.-Y. Lin, J.-Y. Chang, C.-T. Lin, Identification and prediction of dynamic systems using an interactively recurrent self-evolving fuzzy neural network, IEEE Trans. Neural Netw. Learn. Syst. 24 (2013) 310–321.
[41] C.-F. Juang, Y.-Y. Lin, C.-C. Tu, A recurrent self-evolving fuzzy neural network with local feedbacks and its application to dynamic system processing, Fuzzy Sets Syst. 161 (2010) 2552–2568.
[42] S. Ganjefar, M. Tofighi, Single-hidden-layer fuzzy recurrent wavelet neural network: applications to function approximation and system identification, Inf. Sci. 294 (2015) 269–285.
[43] C.-F. Juang, A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms, IEEE Trans. Fuzzy Syst. 10 (2002) 155–170.
[44] H.R. Marateb, S. Goudarzi, A noninvasive method for coronary artery diseases diagnosis using a clinically-interpretable fuzzy rule-based system, J. Res. Med. Sci. 20 (2015).
[45] F. Ornelas-Tellez, E.N. Sanchez, A.G. Loukianov, Discrete-time neural inverse optimal control for nonlinear systems via passivation, IEEE Trans. Neural Netw. Learn. Syst. 23 (2012) 1327–1339.
[46] O. Nelles, Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models, Springer Science & Business Media, 2001.
[47] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Elsevier Science, 2008.
[48] D.G. Luenberger, Y. Ye, Linear and Nonlinear Programming, vol. 116, Springer Science & Business Media, 2008.
[49] J.C. Bezdek, A convergence theorem for the fuzzy ISODATA clustering algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 2 (1980) 1–8.
[50] L. Groll, J. Jakel, A new convergence proof of fuzzy c-means, IEEE Trans. Fuzzy Syst. 13 (2005) 717–720.
[51] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, R. Tibshirani, The Elements of Statistical Learning, vol. 2, Springer, 2009.
[52] J.H. Friedman, Multivariate adaptive regression splines, Ann. Stat. (1991) 1–67.
[53] J. Nocedal, S. Wright, Numerical Optimization, Springer Science & Business Media, 2006.
[54] J.C. Sprott, Elegant Chaos: Algebraically Simple Chaotic Flows, World Scientific, 2010.
[55] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, A. Lendasse, Methodology for long-term prediction of time series, Neurocomputing 70 (2007) 2861–2869.
[56] S. Banerjee, M. Mitra, L. Rondoni, Applications of Chaos and Nonlinear Dynamics in Engineering, vol. 1, Springer Science & Business Media, 2011.
[57] O.E. Rössler, An equation for continuous chaos, Phys. Lett. A 57 (1976) 397–398.
[58] N. Marwan, M.C. Romano, M. Thiel, J. Kurths, Recurrence plots for the analysis of complex systems, Phys. Rep. 438 (2007) 237–329.
[59] F. Takens, Detecting strange attractors in turbulence, in: Dynamical Systems and Turbulence, Warwick 1980, Springer, 1981, pp. 366–381.
[60] A.M. Fraser, H.L. Swinney, Independent coordinates for strange attractors from mutual information, Phys. Rev. A 33 (1986) 1134.
[61] M.B. Kennel, R. Brown, H.D. Abarbanel, Determining embedding dimension for phase-space reconstruction using a geometrical construction, Phys. Rev. A 45 (1992) 3403.
[62] M. Ardalani-Farsa, S. Zolfaghari, Chaotic time series prediction with residual analysis method using hybrid Elman–NARX neural networks, Neurocomputing 73 (2010) 2540–2553.
[63] L.J. Herrera, H. Pomares, I. Rojas, A. Guillén, A. Prieto, O. Valenzuela, Recursive prediction for long term time series forecasting using advanced models, Neurocomputing 70 (2007) 2870–2880.
[64] Y. Ji, J. Hao, N. Reyhani, A. Lendasse, Direct and recursive prediction of time series using mutual information selection, in: Computational Intelligence and Bioinspired Systems, Springer, 2005, pp. 1010–1017.
[65] O.M.M. Mohamed, M. Jaıdane-Saıdane, J. Ezzine, J. Souissi, N. Hizaoui, Variability of predictability of the daily peak load using Lyapunov exponent approach: case of Tunisian power system, in: IEEE Power Tech. Conf., 2007, pp. 1078–1083.
[66] J. Henriques, T. Rocha, Prediction of acute hypotensive episodes using neural network multi-models, in: Computers in Cardiology 2009, 2009, pp. 549–552.