

Identification of Time-Delay Chaotic System with Outliers: Fuzzy Neural Networks Using Hybrid Learning Algorithm

Chia-Nan Ko

Department of Automation Engineering
Nan Kai University of Technology
Nantou, Taiwan, R.O.C.
t105@nkut.edu.tw

Yu-Yi Fu

Department of Automation Engineering
Nan Kai University of Technology
Nantou, Taiwan, R.O.C.
t098@nkut.edu.tw

Guan-Yu Liu

Department of Electronic Engineering
Nan Kai University of Technology
Nantou, Taiwan, R.O.C.
guanyu@nkut.edu.tw

Cheng-Ming Lee

Department of Computer and Communication Engineering
Nan Kai University of Technology
Nantou, Taiwan, R.O.C.
t104@nkut.edu.tw

Abstract—A hybrid learning algorithm is proposed to train fuzzy neural networks (FNNs) for identifying a time-delay chaotic system with outliers. In the proposed algorithm, integrating support vector regression (SVR) and annealing robust time-varying learning algorithm (ARTVLA) to optimize FNNs. In the evolutionary procedure, first, SVR is adopted to determine the number of hidden layer nodes and the initial structure of the FNNs. After initialization, ARTVLA with nonlinear time-varying learning rate is then applied to train FNNs. In ARTVLA, the determination of the learning rate would be an important work for the trade-off between stability and speed of convergence. A computationally efficient optimization method, particle swarm optimization (PSO) method, is adopted to simultaneously find optimal learning rates. Due to the advantages of SVR and ARTVLA (SVR-ARTVLA), the proposed FNNs (SVR-ARTVLA-FNNs) have good performance for identifying a time-delay chaotic system: Mackey Glass system with outliers. Simulation results are illustrated the effectiveness and feasibility of the proposed SVR-ARTVLA-FNNs.

Keywords—fuzzy neural networks; support vector regression; time-delay chaotic system; annealing robust time-varying learning algorithm; particle swarm optimization

I. INTRODUCTION

A chaotic system is a nonlinear deterministic system that has some special features of sensitive dependence on initial conditions and unstable bounded trajectories in the phase space. Control and synchronization of chaotic systems have been investigated in various fields during recent years [1-3]. Due to their characteristics sensitivity to initial conditions, chaotic systems are not easy to identify. Recently, some researchers have endeavored to improve the identification of chaotic time series [4-6].

In the field of artificial intelligence, a fuzzy system using the human-like reasoning and expert knowledge can model complex nonlinear process characterized with ill-defined and uncertain factors. However, a fuzzy system lacks the learning capacity, it is difficult to fine tune fuzzy rules and membership functions from the training data. Neural networks have learning, adaption, and nonlinear mapping capacities, but the mapping rules in the networks is hard to understand. Therefore, FNNs combine the benefits of fuzzy logic and neural network, which is a popular research topic in system identification and control [7-9]. However, the number of fuzzy rule in the simplified fuzzy inference system and the initial parameters of FNNs are difficult to determine.

Recently, SVR [10] incorporated with FNNs has become a popular research topic. In this article, in order to overcome the above problems of training FNNs, first, an SVR method with Gaussian kernel function is adopted to determine the initial structure of the FNNs for identifying nonlinear systems [11, 12]. After initialization, an annealing robust concept in [12, 13] with ARTVLA is then applied to train the FNNs (ARTVLA-FNNs), in which PSO method is adopted to find optimal learning rates during learning procedure. A time-delay chaotic system will be given to illustrate the feasibility and efficiency of the proposed SVR-based ARTVLA-FNNs (SVR-ARTVLA-FNNs) for system identification of the Mackey Glass system.

II. ARCHITECTURE OF FUZZY NEURAL NETWORKS

A. Identification of Chaotic Time series

Let $\{x\}$ be a chaotic system with time delay generated by an m -dimensional nonlinear dynamical system. According to Takens's delay embedding theorem [14], an embedding of

chaotic attractors can be obtained by constructing delay vectors

$$x(t + \tau) = f(x(t), x(t - \tau), x(t - 2\tau), \dots, x(t - (m - 1)\tau)), \quad (1)$$

where $[x(t - \tau), x(t - 2\tau), \dots, x(t - (m - 1)\tau)]$ is a proper vector, τ is the time-delay parameter, and m is the embedding parameter. If m is large enough, there exists a smooth function $f(\cdot)$ [15].

In this paper, the FNN will be applied to identify nonlinear systems due to the advantages of the self-learning ability of neural networks and the adaptive ability of fuzzy logic. The current output can be expressed as

$$\hat{x}_k(t + 1) = \sum_{i=1}^r w_{ik} \exp\left(-\frac{(x - b_i)^2}{2a_i^2}\right) \text{ for } k = 1, 2, \dots, p, \quad (2)$$

where $x = [x(t), x(t - 1), \dots, x(t - n)]$ is the input vector to FNNs, w_{ik} is the synaptic weight of the i th fuzzy rule for the k th FNN output, a_i and b_i are the widths and the centers of the i th membership function for all input data x , respectively.

B. Fuzzy Neural Networks Model

The proposed architecture of FNNs is depicted in Fig. 1. It is a four-layered network structure that consists of input, membership function, inference rule, and output layers. In the FNNs, a wavelet function is adopted as the membership function. Generally, the fuzzy rule of FNNs is expressed as

$$\begin{aligned} R_i : & \text{IF } x_1 \text{ is } A_{i1} \text{ and } \dots \text{ and } x_n \text{ is } A_{in}, \\ \text{THEN } & \hat{y}_{i1} = w_{i1} B_i \text{ and } \dots \text{ and } \hat{y}_{ik} = w_{ik} B_i, \end{aligned} \quad (3)$$

where R_i is the i th fuzzy rule, $1 \leq i \leq r$, x_j is the j th input, $1 \leq j \leq n$, y_{ik} is the k th output contributed by the i th fuzzy rule, $1 \leq k \leq p$, A_{ij} is the membership function for the i th fuzzy rule of the j th input characterized by a wavelet function, w_{ik} is the synaptic weight of the i th inference rule for k th system output, and B_i is the constant consequent of the i th fuzzy rule. The semantic meaning and operation function of the nodes in each layer are described as follows:

Layer 1 (input layer): In this layer, the number of nodes is equal to the number of input variables. These nodes are used to transmit the input linguistic values to the next layer directly.

Layer 2 (membership function layer): In this layer, each node corresponds to one input linguistic term in Layer 1. In other words, the membership value specifying the degree to which an input value belongs to a fuzzy set is calculated in this layer. To describe the linguistic terms, Gaussian function representing fuzzy membership functions is adopted as

$$\mu_{ij}(x_j) = \exp\left(-\frac{(x_j - b_{ij})^2}{2a_{ij}^2}\right), \quad (4)$$

where a_{ij} and b_{ij} for $i = 1, 2, \dots, r$, $j = 1, 2, \dots, n$ are the dilation and the translation of the i th wavelet membership function for the j th input variable x_j , respectively.

Layer 3 (inference rule layer): Each node represents one fuzzy rule in this layer. The AND (min) operation is applied to calculate the output of each node. That is, product inference method is used to determine the firing strength of each node. Each inference rule can be expressed as

$$B_i = \prod_{j=1}^n \mu_{ij}(x_j) \quad \text{for } i = 1, 2, \dots, r, \quad (5)$$

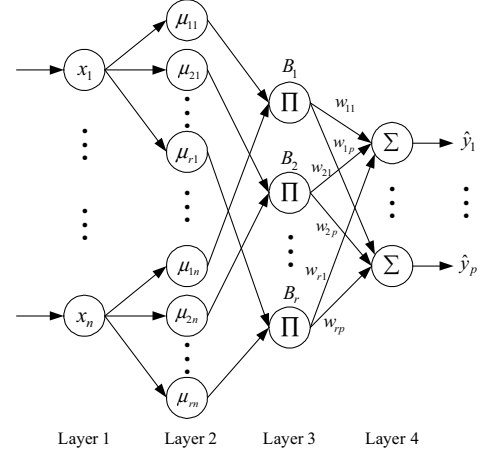


Figure 1. The architecture of fuzzy neural networks (FNNs).

Layer 4 (output layer): In this layer, each node represents an individual output of the FNN by using a simple weighted sum. The mathematic function of the k th output is shown as

$$\hat{y}_k = \sum_{i=1}^r w_{ik} B_i = \sum_{i=1}^r w_{ik} \prod_{j=1}^n \exp\left(-\frac{(x_j - b_{ij})^2}{2a_{ij}^2}\right), \quad (6)$$

where $\hat{y}_k$ for $k = 1, 2, \dots, p$ is the k th FNN output.

It is very difficult, if not impossible, to solve the above problem directly. In usual cases, the initial values of r , w_{ik} , b_{ij} , and a_{ij} are chosen first. Then a training algorithm is applied to FNNs to search for the optimal combination of these values in an iterative manner. However, as mentioned above, there is no way to choose these initial values systematically. Therefore, in the following section, an SVR approach will be proposed to serve for this purpose.

III. SVR-ARTVLA-FNNs BASED ON PSO

A. SVR-Based Initial Structure of FNNs

When approximating an unknown function, SVR method can be adopted to build the initial structure of FNNs [10]. Meanwhile, assume that a set of basis functions, $g_l(x)$, $l = 1, 2, \dots, L$, is given. Then the problem of function

approximation is transformed into finding the parameters of the following basis linear expansion

$$f(\mathbf{x}, \boldsymbol{\theta}) = \sum_{l=1}^L \theta_l g_l(\mathbf{x}) + bias, \quad (7)$$

where $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_L)$ is a parameter vector to be identified and $bias$ is a constant to be determined.

From SVR method, one can find that the solution is to find $f(\mathbf{x}, \boldsymbol{\theta})$ in (7) that minimizes

$$R(\boldsymbol{\theta}) = \frac{1}{N} \sum_{j=1}^N L_{\varepsilon}(y_1^{(j)} - f(\mathbf{x}^{(j)}, \boldsymbol{\theta})), \quad (8)$$

subject to the constraint

$$\|\boldsymbol{\theta}\|^2 < C, \quad (9)$$

where $L_{\varepsilon}(\cdot)$ is the ε -insensitive loss function defined as

$$L_{\varepsilon}(e) = \begin{cases} 0 & \text{for } |e| \leq \varepsilon \\ |e| - \varepsilon & \text{otherwise} \end{cases} \quad (10)$$

for some nonnegative number ε .

Vapnik [10] has applied the kernel function $K(\mathbf{x}_r, \mathbf{x}_s)$ to replace the inner product of basis functions $g_l(\mathbf{x}_r)$ and the Lagrange multiplier method has been adopted to minimize (8). Then the dual optimization problem can be expressed as follows:

$$\begin{aligned} \text{Min } Q(\alpha, \alpha^*) &= \varepsilon \sum_{r=1}^N (\alpha_r + \alpha_r^*) - \sum_{r=1}^N y_r (\alpha_r - \alpha_r^*) \\ &+ \frac{1}{2} \sum_{r,s=1}^N (\alpha_r^* - \alpha_r)(\alpha_s^* - \alpha_s) K(\mathbf{x}_r, \mathbf{x}_s), \end{aligned} \quad (11)$$

subject to the constraint

$$\sum_{r=1}^N \alpha_r^* = \sum_{r=1}^N \alpha_r, \quad 0 < \alpha_r^*, \alpha_r < C \quad (12)$$

and the solution is in the form of the following linear expansion of kernel functions

$$f(\mathbf{x}, \alpha, \alpha^*) = \sum_{l=1}^L (\alpha_l^* - \alpha_l) K(\mathbf{x}, \mathbf{x}_l) + bias. \quad (13)$$

Note that some of $(\alpha_l^* - \alpha_l)$'s are not zeros and the corresponding vectors $\mathbf{x}_l$'s are called *support vectors* (SVs).

The Gaussian function is used as the kernel function, then (13) can be rewritten as

$$f(\mathbf{x}, \lambda) = \sum_{l=1}^{\#SV} \lambda_l \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_l\|^2}{2\sigma_l^2}\right) + bias, \quad (14)$$

where $\#SV$ is the number of SVs, $\lambda_l = (\alpha_l^* - \alpha_l) \neq 0$, and $\mathbf{x}_l$ denotes SVs. Comparing (14) with (2), $\#SV$, l , λ_l , $\mathbf{x}_l$, and σ_l in (14) can be regarded as the r , i , w_{ik} , b_i , and a_i in (2), respectively.

B. ARTVLA-Based FNNs

With an initial structure, a learning algorithm must be established to train FNNs. In the training procedure of the proposed FNNs, the annealing concept in the cost function of robust back-propagation learning algorithm was adopted to overcome the existing problems in robust back-propagation learning algorithm, such as slow convergence rate and getting into local minimum. A cost function for the ARTVLA is defined here as

$$J_k(h) = \frac{1}{N} \sum_{j=1}^N \rho[e_k^{(j)}(h); \beta(h)] \text{ for } k = 1, 2, \dots, p, \quad (15)$$

where

$$\begin{aligned} e_k^{(j)}(h) &= y_k^{(j)} - \hat{f}_k(\mathbf{x}^{(j)}) \\ &= y_k^{(j)} - \sum_{i=1}^L w_{ik} \exp\left(-\frac{\|\mathbf{x}^{(j)} - b_i\|^2}{2a_i^2}\right), \end{aligned} \quad (16)$$

h is the epoch number, $e_k^{(j)}(h)$ is the error between the k th desired output and the k th output of FNNs at epoch h for the k th input-output training data in a nonlinear system, $\beta(h)$ is a deterministic annealing schedule acting like the cut-off point, and $\rho(\cdot)$ is a logistic loss function defined as

$$\rho[e_k^{(j)}; \beta] = \frac{\beta}{2} \ln \left[1 + \frac{(e_k^{(j)})^2}{\beta} \right] \text{ for } k = 1, 2, \dots, p. \quad (17)$$

Based on the gradient-descent kind of learning algorithms, the synaptic weights w_{ik} , the centers b_i , and the widths a_i of Gaussian functions are updated as

$$\Delta w_{ik} = -\eta_{w_i} \frac{\partial J_k}{\partial w_{ik}} = -\frac{\eta_{w_i}}{N} \sum_{j=1}^N \varphi_k(e_k^{(j)}; \beta) \frac{\partial e_k^{(j)}}{\partial w_{ik}}, \quad (18)$$

$$\Delta b_i = -\eta_b \frac{\partial J_k}{\partial b_i} = -\frac{\eta_b}{N} \sum_{j=k}^p \sum_{j=1}^N \varphi_k(e_k^{(j)}; \beta) \frac{\partial e_k^{(j)}}{\partial b_i}, \quad (19)$$

$$\Delta a_i = -\eta_a \frac{\partial J_k}{\partial a_i} = -\frac{\eta_a}{N} \sum_{k=1}^p \sum_{j=1}^N \varphi_k(e_k^{(j)}; \beta) \frac{\partial e_k^{(j)}}{\partial a_i}, \quad (20)$$

$$\varphi_k(e_k^{(j)}; \beta) = \frac{\partial \rho(e_k^{(j)}; \beta)}{\partial e_k^{(j)}} = \frac{e_k^{(j)}}{1 + (e_k^{(j)})^2 / \beta(h)}, \quad (21)$$

where η_{w_k} , η_b , and η_a are the learning rates for the synaptic weight w_{ik} , $k = 1, 2, \dots, p$, the center b_i , and the width a_i , respectively; $\varphi(\cdot)$ is usually called the influence function. In the ARTVLA, the annealing schedule $\beta(h)$ has the properties:

- (A) β_{initial} , $\beta(h)$ for the first epoch, has large values;
- (B) $\beta(h) \rightarrow 0^+$ for $h \rightarrow \infty$;
- (C) $\beta(h) = \kappa/h$ for any h epoch, where κ is a constant.

When the learning rates are constant in (18) to (10), the work for selecting an appropriate learning rate is tedious; moreover, there exists a tendency to get stuck in a near-optimal solution or to converge slowly. To overcome the stagnation in searching a globally optimal solution, an ARTVLA is proposed to approach the optimal solution closely in this paper. In the learning procedure, considering the search space, a large learning rate is selected at the beginning. Once the algorithm is found to converge more and more towards the optimum, the evolution process is gradually converted to a finer and finer tuning by a smaller learning rate in later epochs. Then, in the ARTVLA, a nonlinear time-varying evolution concept is adopted over iterations, in which the learning rates η_w , η_b , and η_a with a high value $\eta_{\max}$ and nonlinearly decreases to $\eta_{\min}$ at the maximal number of epochs, respectively. This means that the mathematical expressions are given as shown as

$$\eta_w = \eta_{\min} + \left(1 - \frac{h}{\text{epoch}_{\max}}\right)^{pw} (\eta_{\max} - \eta_{\min}), \quad (22)$$

$$\eta_b = \eta_{\min} + \left(1 - \frac{h}{\text{epoch}_{\max}}\right)^{pb} (\eta_{\max} - \eta_{\min}), \quad (23)$$

$$\eta_a = \eta_{\min} + \left(1 - \frac{h}{\text{epoch}_{\max}}\right)^{pa} (\eta_{\max} - \eta_{\min}), \quad (24)$$

where $\text{epoch}_{\max}$ is the maximal number of epochs and h is the current number of epochs. In the updated procedure, appropriate functions for the learning rate η_w , η_b , and η_a can promote the performance of FNNs. However, simultaneously determining the optimal combination of pw , pb , and pa is a time-consuming work. An efficient evolutionary algorithm, PSO, will be adopted to determine the optimal solution (pw , pb , pa).

C. Particle Swarm Optimization

PSO is a population-based stochastic searching technique developed by Kennedy and Eberhart [16]. It is similar to the continuous genetic algorithms, in which it begins with a random population matrix and searches for the optimum by updating generations. However, the PSO has no evolution operations such as crossover and mutation. The potential of this technique makes it be a powerful optimization tool which

has been successfully applied to many fields. Nowadays PSO has been developed to be real competitors with other well-established techniques for population-based evolutionary computation [17-19].

In this paper, the PSO method is adopted to find the optimal combination (pw , pb , pa) of learning rates in (22) to (24) for system identification. When applying the PSO method, possible solutions must be encoded into particle positions and a fitness function must be chosen. In the system identification, the goal is to minimize the error between the desired outputs and the trained outputs. Therefore, the root mean squares error ($RMSE$) should be used to define a fitness function. This means that the fitness function will be defined as

$$RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^N (y^{(j)} - \hat{y}^{(j)})^2}, \quad (25)$$

where $y^{(j)}$ is the desired output, $\hat{y}^{(j)}$ is the trained output for N sampling data.

IV. SIMULATION RESULTS

In the identification scheme of a chaotic system, training input-output data are obtained by feeding a time series $[x(t), x(t-\tau), x(t-2\tau), \dots, x(t-(m-1)\tau)]$ to the system and to predict its corresponding output $x(t+\tau)$. Then subject to the same input data, the objective of identification is to construct an SVR-ARTVLA-FNNs model based on PSO method, which produces an output $\hat{x}(t+\tau)$ to approximate $x(t+\tau)$ as closely as possible.

In this section, a time-delay chaotic system with outliers is used to verify the feasibility of the proposed SVR-ARTVLA-FNNs. When applying the proposed algorithm, the population size, the maximal iteration number, and the maximal epoch number are chosen to be 30, 200, and 500, respectively. As for the parameters in PSO, the best ranges were presented [19]. The variables pw , pb and pa in learning rate functions (22) through (24) are all chosen as real numbers in the range $[0.1, 5]$. In PSO, the particles are updated by linearly time-varying evolutionary algorithm in [19]. Meanwhile, the values of $\eta_{\max}$ and $\eta_{\min}$ are set as 1.0 and 0.01, respectively. To evaluate the performance of the proposed FNNs, $RMSE$ of the training data is adopted as

$$RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^N (x_j(t) - \hat{x}_j(t))^2}, \quad (26)$$

where $x_j(t)$ is the desired output of the chaotic time series, $\hat{x}_j(t)$ is the trained output of the proposed FNNs.

Example: A typical chaotic system, Mackey Glass time-delay series, is considered as an example described as follows [6]:

$$x = \frac{px(t-\tau)}{1+x^r(t-\tau)} - qx(t), \quad (27)$$

where $p = 0.2$, $q = 0.1$, $r = 10$, τ is the time-delay parameter. If $\tau \geq 17$, the time series show the chaotic phenomenon. To compare with [6, 20], the time series are chosen as

$$x(t+6) = f(x(t), x(t-6), x(t-12), x(t-18\tau)), \quad (28)$$

where $x(t)$, $x(t-6)$, $x(t-12)$, and $x(t-18)$ are the inputs of the proposed FNNs; $x(t+6)$ is the output of the FNNs.

In the simulation, 1000 time-delay series are generated from (27) with initial value $x(0) = 1.2$. The former 500 data are selected as the training data for identifying the Mackey Glass system shown in Fig. 2. Assume that the membership function is Gaussian function and initial structures are obtained by an ε -SVR approach, in which the parameters are set as $C = 10$, $\varepsilon = 0.3775$ in (9) and (10).

Problem 1:

The training data with no outliers are fed into the FNNs. The initial values of w_{ik} , b_i , and a_i in (2) can be determined by ε -SVR approach. Meanwhile, the number of fuzzy rules is obtained as 23.

Based on the above initial structure of FNNs, ARTVLA with nonlinear learning rates to train the SVR-FNNs, in which the optimal learning rates are determined by PSO method. Then, the optimal sets (pw, pb, pa) in (22) to (24) are obtained as (4.241218, 2.050131, 3.700988). Meanwhile, the final value of $RMSE$ with ARTVLA-FNNs is found to be 0.005886. With the optimal learning rates, the errors of training data after 500 epochs are illustrated in Fig. 3.

Problem 2:

The training data with 10 artificial outliers shown in Fig. 4 are fed into the FNNs. The initial values of w_{ik} , b_i , and a_i in (2) can be determined by ε -SVR method. Meanwhile, the number of fuzzy rules is obtained as 25.

Based on the above initial structure of FNNs, ARTVLA with nonlinear learning rates to train the SVR-FNNs, in which the optimal learning rates are determined by PSO method. Then, the optimal sets (pw, pb, pa) in (22) to (24) are obtained as (2.306907, 0.979595, 2.961404). Meanwhile, the final value of $RMSE$ with ARTVLA-FNNs is found to be 0.007477. With the optimal learning rates, the errors of training data after 500 epochs are illustrated in Fig. 5.

From the above results, the feasibility of the ARTVLA-FNNs has been confirmed. The final $RMSEs$ of the proposed FNNs and other FNNs of [6] are shown in Tables I. From the results, show that the proposed ARTVLA-FNNs can overcome the outliers and get a good performance for training data. Moreover, the comparison of $RMSE$ with other NNs for Mackey-Glass time series without outliers after 500 training epochs is shown in Table II, in which the superiority of the proposed ARTVLA-FNNs can be verified.

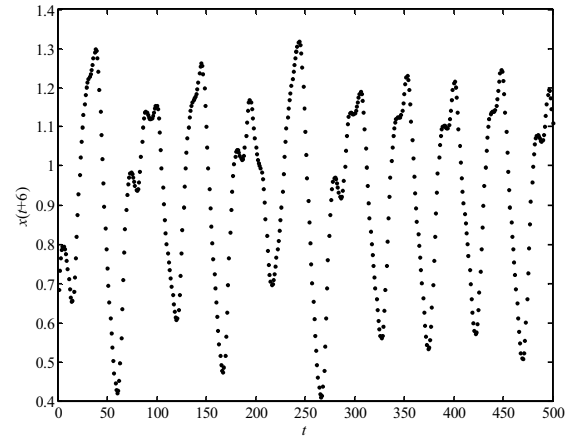


Figure 2. Training data without outliers for identifying time-delay Mackey Glass system.

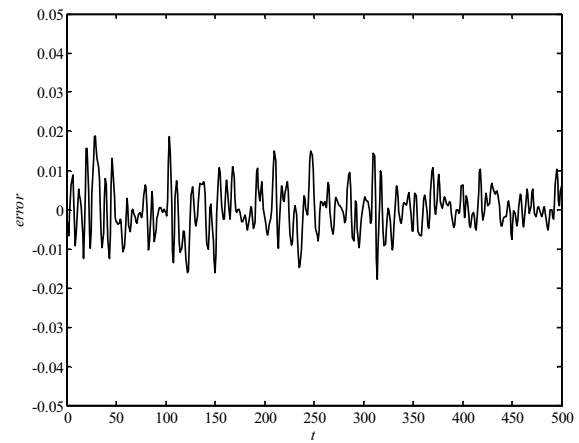


Figure 3. The plots of $error = x(t) - \hat{x}(t)$ for 500 training data without outliers after 500 training epoch.

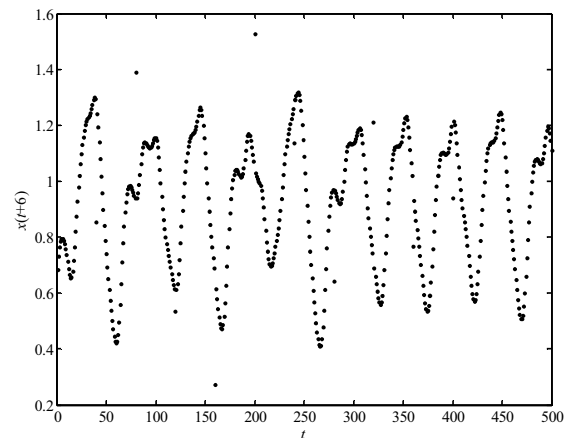


Figure 4. Training data with 10 artificial outliers for identifying time-delay Mackey Glass system.

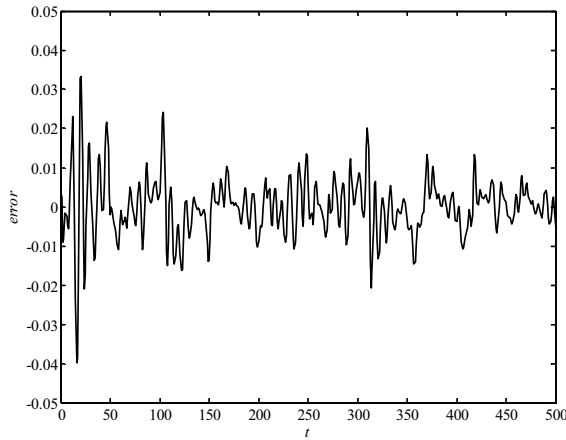


Figure 5. The plots of $error = x(t) - \hat{x}(t)$ for 500 training data with 10 artificial outliers after 500 training epoch.

TABLE I. COMPARISON OF *RMSE* USING VARIOUS METHODS FOR MACKEY-GLASS TIME-SERIES WITH ARTIFICIAL OUTLIERS.

ARTVLA-FNNs	ARFNNs [6]	ANFIS	RBFNs
0.0075	0.0091	0.0201	0.0204

TABLE II. COMPARISON OF *RMSE* USING VARIOUS METHODS FOR MACKEY-GLASS TIME-SERIES WITHOUT OUTLIERS.

Method	Training data	<i>RMSE</i>
ANFIS	500	0.0837
Gu and Wang [20]	500	0.1095
ARFNNs [6]	500	0.0073
ARTVLA-FNNs	500	0.0059

V. CONCLUSIONS

This paper presents the integrating SVR and ARTVLA to train FNNs for identification of a time-delay chaotic system. With the initial structure of the FNNs using SVR method, the nonlinear time-varying learning rates are simultaneously determined by PSO method to perform the ARTVLA. Then the optimal FNNs are obtained to identify the Mackey-Glass chaotic system. From the simulation results, one can conclude that the proposed SVR-ARTVLA-FNNs using PSO method have good performance for identification of the time-delay series system. Meanwhile, the superiority of the proposed ARTVLA-FNNs with nonlinear learning rates over the other NNs for identification has been verified. The proposed SVR-ARTVLA-FNNs can be further extended to identify more complex chaotic systems.

ACKNOWLEDGMENT

This work was supported in part by the National Science Council, Taiwan, R.O.C., under grants NSC 99-2221-E-252-017.

REFERENCES

- [1] J. H. Kim, C. W. Park, E. Kim, and M. Park, "Fuzzy adaptive synchronization of uncertain chaotic systems," *Physics Letters A*, vol. 334, pp. 295–305, 2005.
- [2] X. Wu, Z. H. Guan, Z. Wu, and T. Li, "Chaos synchronization between Chen system and Genesio system," *Physics Letters A*, vol. 364, pp. 484–487, 2007.
- [3] J. H. Park, "Adaptive control for modified projective synchronization of a four-dimensional chaotic system with uncertain parameters," *Journal of Computational and Applied Mathematics*, vol. 213, pp. 288–293, 2008.
- [4] H. Gu and H. W. Wang, "Fuzzy prediction of chaotic time series based on singular value decomposition," *Applied Mathematics and Computation*, vol. 185, pp. 1171–1185, 2007.
- [5] S. T. Pan and C. C. Lai, "Identification of chaotic systems by neural network with hybrid learning algorithm," *Chaos, Solitons and Fractals*, vol. 37, no. 1, pp. 233–244, 2008.
- [6] Y. Y. Fu, C. J. Wu, J. T. Jeng, and C. N. Ko, "ARFNNs with SVR for prediction of chaotic time series with outliers," *Expert Systems with Applications*, vol. 37, pp. 4441–4451, 2010.
- [7] C. F. Juang and C. T. Lin, "A recurrent self-organizing neural fuzzy inference network," *IEEE Transactions on Neural Networks*, vol. 10, no. 4, pp. 828–845, 1999.
- [8] S. Abbasbandy and M. Otadi, "Numerical solution of fuzzy polynomials by fuzzy neural network," *Applied Mathematics and Computation*, vol. 181, no. 2, pp. 1084–1089, 2006.
- [9] A. Azadeh, S. F. Ghaderi, M. Anvari, M. Saberi, and H. Izadbakhsh, "An integrated artificial neural network and fuzzy clustering algorithm for performance assessment of decision making units," *Applied Mathematics and Computation*, vol. 187, no. 2, pp. 584–599, 2007.
- [10] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
- [11] S. Gao, H. W. Dai, Y. Y. Zhu, and Z. Tang, "Annealing chaotic pattern search learning method for multi-layer neural networks," *International Journal of Computer Science and Network Security*, vol. 7, no. 6, pp. 137–145, 2007.
- [12] C. C. Chuang, S. F. Su, and C. C. Hsiao, "The annealing robust backpropagation (BP) learning algorithm," *IEEE Transactions on Neural Networks*, vol. 11, no. 5, pp. 1067–1077, 2000.
- [13] C. C. Chuang, J. T. Jeng, and P. T. Lin, "Annealing robust radial basis function networks for function approximation with outliers," *Neurocomputing*, vol. 56, pp. 123–139, 2004.
- [14] F. Takens, *Detecting strange attractors in turbulence. Dynamical systems and turbulence*. Berlin: Springer, pp. 336–381, 1981.
- [15] L. Cao, A. Mees, and K. Judd, "Dynamics from multivariate time series," *Physica D*, vol. 121, pp. 75–88, 1998.
- [16] J. Kennedy and R. Eberhart, "Particle swarm optimization," In *Proceeding of the IEEE International Conference on Neural Networks*, Washington, DC, USA, vol. 4, pp. 1942–1948, 1995.
- [17] D. W. Boeringer and D. H. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 3, pp. 771–779, 2004.
- [18] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Advanced Engineering Informatics*, vol. 19, no. 1, pp. 43–53, 2005.
- [19] A. Ratnaweera, S. K. Halgamuge, and C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.
- [20] H. Gu and H. Wang, "Fuzzy prediction of chaotic time series based on singular value decomposition," *Applied Mathematics and Computation*, vol. 185, pp. 1171–1185, 2007.