Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks

Kot, Alex Chichung; Patra, Jagdish Chandra

2002

Patra, J. C., & Kot, A. C. (2002). Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 32(4), 505-511.

https://hdl.handle.net/10356/94174

https://doi.org/10.1109/TSMCB.2002.1018769

Nonlinear Dynamic System Identification Using Chebyshev Functional Link Artificial Neural Networks

Jagdish C. Patra, Member, IEEE, and Alex C. Kot, Senior Member, IEEE

Nanyang Technological University, Singapore 639 798
E-mail: aspatra@ntu.edu.sg; eackot@ntu.edu.sg.

Abstract

A computationally efficient artificial neural network (ANN) for the purpose of dynamic nonlinear system identification is proposed. The major drawback of feedforward neural networks such as a multilayer perceptron (MLP) trained with backpropagation (BP) algorithm is that it requires a large amount of computation for learning. We propose a single-layer functional link ANN (FLANN) in which the need of hidden layer is eliminated by expanding the input pattern by Chebyshev polynomials. The novelty of this network is that it requires much less computation than that of a MLP. We have shown its effectiveness in the problem of nonlinear dynamic system identification. In presence of additive Gaussian noise to the plant, the performance of the proposed network is found similar or superior to that of a MLP. Performance comparison in terms of computational complexity has also been carried out.

Index Terms—Chebyshev polynomials, functional link neural networks, multilayer perception, nonlinear system identification.

I. INTRODUCTION

IDENTIFICATION of a complex dynamic plant is a major concern in control theory. This is due to the fact that effective solutions are needed for some of the long-standing problems of automatic control, such as: to work with more and more complex systems, to satisfy stricter design criteria and to fulfill the previous two points with less *a priori* knowledge of the plant [1]. In this context, great efforts are being made in the area of system identification, toward development of nonlinear dynamic models of real processes.

Recently, artificial neural networks (ANNs) have emerged as a powerful learning technique to perform complex tasks in highly nonlinear dynamic environments [2]. Some of the prime advantages of using ANN models are: their ability to learn based on optimization of an appropriate error function and their excellent performance for approximation of nonlinear functions. There are different paradigms of ANNs proposed by different researchers for the task of system identification and control. Recently, recurrent ANNs with internal dynamics have been proposed with adequate results [3], [4]. These networks are capable of effective identification and control of complex process dynamics, but with the expense of large computational complexity. A continuous-time additive dynamic neural network has been proposed to identify real processes using on-line training methods [5]. The models obtained with this approach are in state-space and work quite effectively in continuous-time domain.

Presently, most of the ANN-based system identification techniques are based on multilayer feedforward networks such as multilayer perceptron (MLP) trained with backpropagation (BP) or more efficient variation of this algorithm [6]–[9]. This is due to the fact that these networks are robust and effective in modeling and control of complex dynamic plants. Narendra and Parthasarathy [6] have proposed effective identification and control of dynamic systems using MLP networks. These methods have been applied successfully to several real processes for example: control of truck-backer-upper problem [7] and robot arm control [8].

As an alternative to the MLP, there has been considerable interest in radial basis function (RBF) networks [10]–[13], primarily because of its simpler structure. The RBF networks can learn functions with local variations and discontinuities effectively and also possess universal approximation capability [13]. This network represents a function of interest by using members of a family of compactly or locally supported basis functions, among which radially-symmetric Gaussian functions are found to be quite popular. A RBF network has been proposed for effective identification of nonlinear dynamic systems [14], [15]. In these networks, however, choosing an appropriate set of RBF centers for effective learning, still remains as a problem. Considering as a special case of RBF networks, the use of wavelets in neural networks have been proposed [16], [17]. In these networks, the radial basis functions are replaced by wavelets which are not necessarily radial-symmetric. Wavelet neural networks for function learning and nonparametric estimation can be found in [18], [19].

The functional link ANN (FLANN) proposed by Pao [20] can be used for function approximation and pattern classification with faster convergence and lesser computational complexity than a MLP network. A single-layer orthonormal neural network using Legendre polynomials has been reported for static function approximation [21]. Sadegh [22] reported a functional basis perceptron network for functional identification and control of nonlinear systems. Linear and nonlinear ARMA model parameter estimation using an ANN with polynomial activation functions for biomedical application has been reported [23]. A FLANN approach using a tensor model for expansion has been applied to thermal dynamic system identification [24]. A FLANN using sin and cos functions for functional expansion for the problem of nonlinear dynamic system identification has been reported [25]. It is shown that with proper choice of functional expansion, the FLANN is capable of performing as good as and in some cases, even better than MLP in the system identification problem. However, input plant noise which is inherent in practical systems was not considered.

Pattern classification using Chebyshev neural networks has been reported in [26]. However, its performance compared to an MLP has not been studied. A Chebyshev polynomial-based unified model ANN for static function approximation is reported [27]. It is based on a FLANN with Chebyshev polynomial expansion in which recursive least square learning algorithm is used. It is pointed out that this network has universal approximation capability and has faster convergence than a MLP network.

One of the solutions for the problem of slow convergence of MLP is to use some efficient learning algorithm instead of BP algorithm. In this direction, the scaled complex conjugate gradient algorithm as proposed by Moeller [28] is of great importance. This algorithm chooses the search direction and the step size using information from a second order Taylor expansion of the error function. Some of the other proposals on higher order conjugate gradient algorithms can be found in [29]–[31]. For data classification and function interpolation problems a MLP trained by conjugate gradient algorithm has been reported [32].

In this paper, we propose a FLANN structure similar to [27] for the problem of identification of nonlinear dynamic systems in presence of input plant noise. Generally, a linear node in its output is used in the FLANN structure reported by other researchers. But, in our proposed network, we have used a nonlinear node with nonlinearity in the output layer for better performance. In [27], identification of only static systems without any consideration to plant noise has been reported. For functional expansion of the input pattern, we have chosen the Chebyshev polynomials and the network is named as Chebyshev-FLANN (CFLANN). Selecting some of the system examples reported by Narendra and Parthasarathy [6], we have compared performance of the proposed network with that of a MLP network used by them. The primary purpose of this paper is to highlight effectiveness of the proposed simple ANN architecture in the problem of nonlinear dynamic system identification in presence of additive plant noise.

II. CHARACTERIZATION AND IDENTIFICATION OF SYSTEMS

The primary concern in the problem of characterization is the mathematical representation of the system under study. Let us express the model of a system by an operator **P** from an input space $\mathcal{U}$ into an output space $\mathcal{Y}$. The objective is to categorize the class $\mathcal{P}$ to which **P** belongs. For a given class $\mathcal{P}$, $\mathbf{P} \in \mathcal{P}$, the identification problem is to determine a class $\hat{\mathcal{P}} \subset \mathcal{P}$ and $\hat{\mathbf{P}} \in \hat{\mathcal{P}}$ such that $\widehat{\mathbf{P}}$ approximates **P** in some desired sense. In a static system, the space $\mathcal{U}$ and $\mathcal{Y}$ are subsets of $R^n$and R^m, respectively. Whereas, in a dynamic system, they are assumed to be bounded Lebesgue integrable functions in the interval$[0, T]$ or $[0, \infty]$. However, in both cases, the operator **P** is defined implicitly by the specified input-output pairs [6].

A typical example of identification of static system is the problem of pattern recognition. By a decision function **P**, compact input sets $U_i \subset R^n$are mapped into elements $y_i \in R^m$ for $i = 1,2, \ldots$ in the output space. The elements U_i of denote the pattern vectors corresponding to class y_i. Whereas, in a dynamic system, the input–output pairs of the time function $\{u(t), y(t)\}$, $t \in [0, T]$, implicitly define the operator **P** describing the dynamic plant. The main objective in both types of identification is to determine $\widehat{\mathbf{P}}$ such that

$$\|y - \hat{y}\| = \|\mathbf{P}(u) - \hat{\mathbf{P}}(u)\| < \epsilon \qquad (1)$$

where $u \in \mathcal{U}, \epsilon$ is some desired small value > 0 and $\|.\|$ is a defined norm on the output space. In (1), $\widehat{\mathbf{P}}$ and $\mathbf{P}$ denote the output of the identified model and the plant, respectively. The error $e = y - \hat{y}$ is the difference between the observed plant output and the output generated by the model.

In Fig. 1, a schematic diagram of system identification of a time-invariant, causal, discrete-time plant is shown. The input and output of the plant are represented by u and $\mathbf{p}(u)$, respectively, where u is assumed to be an uniformly bounded function of time. The stability of the plant is assumed with a known parameterization but with unknown parameter values. The objective is to construct a suitable model generating an output $\widehat{\mathbf{P}}(u)$ which approximates the plant output $\mathbf{P}(u)$. In the present study we considered single-input–single-output (SISO) plants and four models are introduced as follows.

- Model 1:

$$y_p(k+1) = \sum_{i=0}^{n-1} \alpha_i y_p(k-i) + g\left[u(k), u(k-1), \ldots, u(k-m+1)\right].$$

- Model 2:

$$y_p(k+1) = f\left[y_p(k), y_p(k-1), \ldots, y_p(k-n+1)\right] + \sum_{i=0}^{m-1} \beta_i u(k-i).$$

- Model 3:

$$y_p(k+1) = f\left[y_p(k), y_p(k-1), \ldots, y_p(k-n+1)\right] + g\left[u(k), u(k-1), \ldots, u(k-m+1)\right].$$

- Model 4:

$$y_p(k+1) = f\left[y_p(k), y_p(k-1), \ldots, y_p(k-n+1); +u(k), u(k-1), \ldots, u(k-m+1)\right].$$

Here $u(k)$ and $y_p(k)$ represent the input and the output of the SISO plant at the kth time instant, respectively and $m \leq n$. In this study ANNs (MLP and CFLANN) have been used to construct the nonlinear functions f and g so as to approximate such mappings over compact sets. It is assumed that the plant is bounded-input–bounded-output (BIBO) stable. In contrast to this, the stability of ANN model can not be assured. Therefore, in order to guarantee that the parameters of the ANN model converge, a series–parallel scheme is utilized. In this scheme, output of the plant instead of the ANN model is fed back into the model during training of the ANN [6].

III. THE ARTIFICIAL NEURAL NETWORKS

Here, we briefly describe the architecture and learning algorithm for the two types of ANNs (*i.e.*, MLP and FLANN) used in this study.

A. Multilayer Perceptron

The MLP is a multilayer architecture with one or more hidden layer(s) between its input and output layers. All the nodes of a lower layer are connected with all the nodes of the adjacent layer through a set of weights. All the nodes in all layers (except the input layer) of the MLP contain a nonlinear $\tanh()$ function. A pattern is applied to the input layer, but no computation takes place in this layer. Thus, the output of the nodes of this layer is the input pattern itself. The weighted sum of outputs of a lower layer is passed through the nonlinear function of a node in the upper layer to produce its output. Thus, the outputs of all the nodes of the network are computed. The outputs of the final layer (output layer) are compared with a target pattern associated with the input pattern. The error between the target pattern and the output layer node is used to update the weights of the network. The mean square error (MSE) is used as a cost function. The BP algorithm attempts to minimize this cost function by adapting all weights of the network. More details about the MLP and BP algorithm can be found in [2].

B. Functional Link ANN

The FLANN, initially proposed by Pao [20], is a single-layer ANN structure capable of forming complex decision regions by generating nonlinear decision boundaries. In a FLANN, the need of hidden layer is removed. In contrast to linear weighting of the input pattern produced by the linear links of a MLP, the functional link acts on an element or the entire pattern itself by generating a set of linearly independent functions. In this study, the functional expansion block comprises of a subset of Chebyshev polynomials. Separability of the input patterns in the enhanced pattern space is possible. For example, consider a 2-D input pattern $X = [x_1 \;\; x_2]^T$. An enhanced pattern obtained by using Chebyshev functions is given by $X^* = [1 \;\, x_1 \;\, T_2(x_1) \dots x_2 \;\, T_2(x_2) \dots]^T$. This enhanced pattern can be used for classification/estimation purposes. The BP algorithm used to train the FLANN becomes simpler and has a faster convergence due to its single layer architecture. A generalized FLANN structure with a single output node is shown in Fig. 2.

C. Learning Algorithm for FLANN

Learning of an ANN may be considered as approximating or interpolating a continuous, multivariate function $f(X)$ by an approximating function $f_W(X)$. In the FLANN, a set of basis functions Φ and a fixed number of weight parameters W are used to represent $f_W(X)$. With a specific choice of a set of functions, the problem is then to find the weight parameters W that provides the best possible approximation of f on the set of input–output examples. This can be achieved by recursively updating W. Detailed theory on the FLANN may be found in [22].

Let a training pattern be denoted by $\{X_k, Y_k\}$ and the weight matrix by $W(k)$. Discrete time index, k is given by $k = \kappa + \lambda K$ for $\kappa = 1,2, \dots K$ and $\lambda = 0,1,2, \dots$, where K is total number of training patterns. At kth instant, the n-dimensional input pattern and the m-dimensional FLANN output are given by $X_k = [\, x_1(k) \;\; x_2(k) \;\; \dots \;\; x_n(k)]^T$ and $\hat{Y}_k = [\, \hat{y}_1(k) \;\; \hat{y}_2(k) \;\; \dots \;\; \hat{y}_m(k)]^T$, respectively. Its corresponding target pattern is represented by $Y_k = [\, y_1(k) \;\; y_2(k) \;\; \dots \;\; y_m(k)]^T$. The dimension of the input pattern increases from n to N by a basis function Φ given by $\Phi(X_k) = [\phi_1(X_k) \;\; \phi_2(X_k) \;\; \dots \;\; \phi_N(X_k)]^T$. The $(m \times N)$ -dimensional weight matrix is given by $W(k) = [W_1(k) \;\; W_2(k) \;\; \dots \;\; W_m(k)]^T$ where, $W_j(k)$ is the weight vector associated with jth output and is given by $W_j(k) = [w_{j1}(k) \;\; w_{j2}(k) \;\; \dots \;\; w_{jN}(k)]$. The jth output of the FLANN is given by

$$\begin{aligned} \hat{y}_j(k) &= \rho\left(\sum_{i=1}^{N} w_{ji}(k)\phi_i(X_k)\right) \\ &= \rho\left(W_j(k)\Phi(X_k)\right) \end{aligned} \tag{2}$$

for $j = 1,2, \dots, m$. The error associated with jth output mode is given by $e_j(k) = y_j(k) - \hat{y}_j(k)$. Using the BP algorithm, weights of the FLANN can be updated as

$$\begin{aligned} W(k+1) &= W(k) + \mu\Delta(k) + \gamma\Delta(k-1) \\ \Delta(k) &= \delta(k)\left[\Phi(X_k)\right]^T \end{aligned} \tag{3}$$

where, $\delta(k) = [\delta_1(k) \;\; \delta_2(k) \;\; \dots \;\; \delta_m(k)]^T$, $\delta_j(k) = (1 - \hat{y}_j^2(k))e_j(k)$ and μ and γ are learning parameters and momentum factor, respectively.

D. Chebyshev Expansion

In this study we used Chebyshev polynomials for functional expansion as shown in Fig. 2. These polynomials are easier to compute than that of trigonometric polynomials. In our study, we found superior performance by using CFLANN. The first few Chebyshev polynomials are given by: $T_o(x) = 1.0$, $T_1(x) = x$ and $T_2(x) = 2x^2 - 1$ .The higher order Chebyshev polynomials may be generated by the recursive formula given by

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x). \tag{4}$$

E. Computational Complexity

Here we present a comparison of computational complexity between the MLP and FLANN structures trained by BP algorithm. Let us consider an L-layer MLP with n_l nodes (excluding the threshold unit) in layer l, $l = 0,1, \dots, L$, where n_o and n_L represent number of nodes in the input and output layers, respectively. An L-layer ANN architecture may be represented by $\{n_o - n_1 - \cdots - n_{L-1} - n_L\}$. Three basic computations, *i.e.*, addition, multiplication and computation of $\tanh(.)$ are involved for updating weights of the ANN. The computations in the network are due to the following requirements:

1) forward calculations to find the activation value of all the nodes of the entire network;
2) back-error propagation for calculation of square error derivatives;
3) updating weights of the entire network.

Total number of weights to be updated in one iteration in an MLP is given by $(\sum_{l=0}^{L-1}(n_l + 1)n_{l+1})$, whereas in the case of a FLANN it is only $(n_o + 1)n_1$. A comparison of the computational requirements in one iteration of training using BP algorithm for the two ANNs are provided in Table I. From this table it may be seen that the number of additions, multiplications and computation of are much less in the case of a FLANN than that of a MLP network. As the number of hidden layers and number of nodes in a layer increase, the computations in a MLP increase. However, due to absence of hidden layer in the FLANN, its computational complexity reduces drastically.

IV. SIMULATION STUDIES

Extensive simulation studies were carried out with several examples of nonlinear dynamic systems. We compared performance of the proposed CFLANN with that of a MLP for those problems reported by Narendra and Parthasarathy [6]. For this purpose, we used the same MLP architecture $\{1-20-10-1\}$ as used by them.

During the training phase, an uniformly distributed random signal over the interval $[-1,1]$ was applied to the plant and the ANN model. White Gaussian noise was added to the input of the plant. As it is usually done in adaptive algorithms, the learning parameter μ and the momentum factor γ in both ANNs were chosen after several trials to obtain best results. In a similar manner, the functional expansion of the FLANN was carried out. The adaptation continued for 50000 iterations during which the series–parallel identification scheme was used. Thereafter, adaption was discontinued and the ANN was used for identification purpose. During the test phase, the effectiveness of the ANN models were studied by presenting a sinusoidal signal given by

$$\begin{aligned} u(k) &= \sin\left(\frac{2\pi k}{250}\right) \text{ for } 0 < k \leq 250, \\ &= 0.8\sin\left(\frac{2\pi k}{250}\right) + 0.2\sin\left(\frac{2\pi k}{25}\right) \text{ for } k > 250. \quad (5) \end{aligned}$$

Performance comparison between the MLP and the CFLANN was carried out in terms of output estimated by the ANN model, actual output of the plant and modeling error. A standard quantitative measure for performance evaluation is the normalized mean square error (NMSE) and is defined as [33]

$$\Gamma = \frac{1}{\sigma^2 T_D} \sum_{k=1}^{T_D} [y(k) - \hat{y}(k)]^2 \qquad (6)$$

where $y(k)$ and $\hat{y}(k)$ represent plant ANN model outputs at k th discrete time, respectively and σ^2 denotes variance of the plant output sequence over the test duration T_D. It may be noted that, in the results for all the examples provided below, the ANN model was trained with random signals. Whereas, testing of the ANNs were carried out by applying a sinusoidal signal (5) to the plant and the model. The results are shown for 600 discrete samples, *i.e.*, $T_D = 600$.

Example 1: We consider a system described by the difference equation of Model 1. The plant is assumed to be of second order and is described by the following difference equation:

$$y_p(k+1) = 0.3y_p(k) + 0.6y_p(k-1) + g\left[u(k)\right] \qquad (7)$$

where nonlinear function g is unknown, but $\alpha_o = 0.3$ and $\alpha_1 = 0.6$ are assumed to be known. The unknown function g is given by: $g(u) = 0.6\sin(\pi u) + 0.3\sin(3\pi u) + 0.1\sin(5\pi u)$. To identify the plant, a series-parallel model was considered which is governed by the following difference equation:

$$\hat{y}_p(k+1) = 0.3y_p(k) + 0.6y_p(k-1) + \mathcal{N}\left[u(k)\right]. \qquad (8)$$

The MLP used for the purpose of identification has a structure of $\{1-20-10-1\}$. For the CFLANN, the input was expanded to 14 terms using Chebyshev polynomials. Both μ and γ [refer (3)] were chosen to be 0.5 in the two ANNs and a white Gaussian noise of -10 dB was added to the input of the plant. The results of the identification with the sinusoidal signal (5) are shown in Fig. 3. It may be seen from this figure that the identification of the plant is satisfactory for both the ANNs. But, the estimation error in the CFLANN is found to be less than that of the MLP. The NMSE for the MLP and the CFLANN models are found to be -16.69 dB and -26.22 dB, respectively.

Example 2: We consider a plant described by the difference equation of Model 2:

$$y_p(k+1) = f\left[y_p(k), y_p(k-1)\right] + u(k). \qquad (9)$$

It is known a *priori* that the output of the plant depends only on the past two values of the output and the input of the plant. The unknown function f is given by $f(y_1, y_2) = y_1 y_2 (y_1 + 0.5)(y_1 - 1.0)/(1.0 + y_1^2 + y_2^2)$. The series-parallel scheme used to identify the plant is described by

$$\hat{y}_p(k+1) = \mathcal{N}\left[y_p(k),\, y_p(k-1)\right] + u(k). \qquad (10)$$

A MLP of $\{1-20-10-1\}$ structure was used. In the CFLANN, the 2-dimensional input vector was expanded by the Chebyshev polynomials upto 12 terms. A Gaussian noise of -30 dB was added to the input of the plant. For the MLP the values of μ and γ were set to 0.05 and 0.10, respectively. In the case of CFLANN, the values of μ and γ

were chosen to be 0.07 and 0.01, respectively. After the completion of training, the sinusoidal signal (5) was applied to the plant and the ANN models. The results of the identification are shown in Fig.4. The values of NMSE for the MLP and the CFLANN are found to be -19.47 dB and -21.20 dB, respectively. It may be seen that the performance of both ANN models are similar and satisfactory.

Example 3: Here, the plant is of Model 3 and is described by the following difference equation:

$$y_p(k+1) = f\left[y_p(k)\right] + g\left[u(k)\right] \qquad (11)$$

where the unknown functions f and g have are given by: $f(y) = (y(y+0.3))/(1.0 + y^2)$ and $g(u) = u(u+0.8)(u-0.5)$. The series parallel model for identification is given by

$$\hat{y}_p(k+1) = \mathcal{N}_1\left[y_p(k)\right] + \mathcal{N}_2\left[u(k)\right] \qquad (12)$$

where $\mathcal{N}_1$ and $\mathcal{N}_2$ are the two ANNs used to approximate the two nonlinear functions f and g ,respectively.

In the case of MLP, both $\mathcal{N}_1$ and $\mathcal{N}_2$ were represented by $\{1-20-10-1\}$whereas, in the case of CFLANN these were represented by $\{14-1\}$ structure. To improve the learning process, the output of the plant was scaled down by a scale factor (SF) before applying it to the ANN model. The SF was chosen as 2.0. The learning parameters μ and γ were chosen as 0.50 and 0.25, respectively, for the MLP. Whereas, in the case of CFLANN, μ and γ were selected as 0.20 and 0.10, respectively. A Gaussian noise of -20 dB was added to the input of the plant. The results of the identification are depicted in Fig. 5. The NMSE values are found to be -19.45 dB and -20.25 dB for the MLP and CFLANN models, respectively. It may be seen that the CFLANN is capable of estimating the plant response similar to that of the MLP.

Example 4: The plant model selected here is the most general of all the examples chosen. It belongs to the Model 4 and is described by

$$y_p(k+1) = f\left[y_p(k), y_p(k-1), y_p(k-2), u(k), u(k-1)\right] \qquad (13)$$

where the unknown function f is given by $f[a_1, a_2, a_3, a_4, a_5] = a_1 a_2 a_3 a_5 (a_3 - 1.0)/(1.0 + a_2^2 + a_3^2)$. The series-parallel model for identification of the plant is given by

$$\hat{y}_p(k+1) = \mathcal{N}\left[y_p(k), y_p(k-1), y_p(k-2), u(k), u(k-1)\right]. \qquad (14)$$

In the case of MLP and FLANN, $\mathcal{N}$ is represented by $\{5-20-10-1\}$ and $\{10-1\}$ structures, respectively. The inputs, y_p's and u's were expanded by using the Chebyshev polynomials to 10 terms and used in the CFLANN for identification of the plant. A Gaussian noise of -10 dB was added to input of the plant and a SF of 1.5 was selected. The learning parameter μ and the momentum factor γ for the MLP model were chosen as

0.01 and 0.10, respectively. Whereas, both μ and γ for the CFLANN were selected as 0.50.

The outputs of the plant and the ANN models along with their corresponding errors are shown in Fig. 6. The NMSE for the MLP and CFLANN models were found to be -18.43 dB and -16.60 dB, respectively. It may be observed that the performance of the CFLANN is slightly inferior to that of the MLP model.

Comparison of computational load between a MLP and a CFLANN for the four examples studied is provided in Table II. It may be seen that computational requirements of a CFLANN in terms of number of additions, multiplications and computation of $\tanh$ are much lower than that of a MLP [34], [35].

V. CONCLUSIONS

We have proposed a novel single-layer ANN structure for identification of nonlinear dynamic systems. In a functional-link ANN, functional expansion of the input increases the dimension of the input pattern. Thus, creation of nonlinear decision boundaries in the multidimensional input space and identification of complex nonlinear dynamic systems become easier. In the proposed CFLANN, the input functional expansion is carried out using the Chebyshev polynomials. In the four models of nonlinear dynamic systems considered in this study, the CFLANN is found to be effective in identification of all the systems. The prime advantage of the proposed ANN is its reduced computational complexity without any sacrifice on its performance. Simulation results indicate that performance of the proposed network is as good as that of a MLP network in presence of additive noise to the system. The CFLANN may be used for on-line signal processing applications due to its less computational requirement and satisfactory performance.

REFERENCES

[1] P. J. Antsaklis, “Neural networks in control systems,” *IEEE Contr. Syst. Mag.*, pp. 3–5, Apr. 1990.

[2] S. Haykin, *Neural Networks*. Ottawa, ON, Canada: Maxwell Macmillan, 1994.

[3] P. S. Sastry, G. Santharam, and K. P. Unnikrishnan, “Memory neural networks for identification and control of dynamical systems,” *IEEE Trans. Neural Networks*, vol. 5, pp. 306–319, Mar. 1994.

[4] A. G. Parlos, K. T. Chong, and A. F. Atiya, “Application of recurrent multilayer perceptron in modeling of complex process dynamics,” *IEEE Trans. Neural Networks*, vol. 5, pp. 255–266, Mar. 1994.

[5] R. Grino, G. Cembrano, and C. Torras, “Nonlinear system identification using additive dynamic neural networks- two on-line approaches,” *IEEE Trans. Circuits Syst.-I*, vol. 47, pp. 150–165, Feb. 2000.

[6] K. S. Narendra and K. Parthasarathy, “Identification and control of dynamical systems using neural networks,” *IEEE Trans. Neural Networks*, vol. 1, pp. 4–26, Jan. 1990.

[7] D. H. Nguyen and B. Widrow, “Neural networks for self-learning control system,” *Int. J. Contr.*, vol. 54, no. 6, pp. 1439–1451, 1991.

[8] G. Cembrano, G. Wells, J. Sarda, and A. Ruggeri, “Dynamic control of a robot arm based on neural networks,” *Contr. Eng. Practice*, vol. 5, no. 4, pp. 485–492, 1997.

[9] S. Lu and T. Basar, “Robust nonlinear system identification using neural network models,” *IEEE Trans. Neural Networks*, vol. 9, pp. 407–429, May 1998.

[10] T. Poggio and F. Girosi, “Networks for approximation and learning,” *Proc. IEEE*, vol. 78, pp. 1481–1497, Sep. 1990.

[11] J. Moody and C. J. Darken, “Fast learning in networks of locally-tuned processing units,” *Neural Comput.*, vol. 1, pp. 281–294, 1989.

[12] J. Park and I. W. Sandberg, “Universal approximation using radial basis function networks,” *Neural Comput.*, vol. 3, pp. 246–257, 1991.

[13] E. J. Hartman, J. D. Keeler, and J. M. Kowalski, “Layered neural networks with Gaussian hidden units as universal approximation,” *Neural Comput.*, vol. 2, pp. 210–215, 1990.

[14] S. Chen, S. A. Billings, and P. M. Grant, “Recursive hybrid algorithm for nonlinear system identification using radial basis function networks,” *Int. J. Contr.*, vol. 55, no. 5, pp. 1051–1070, 1992.

[15] S. V. T. Elanayar and Y. C. Shin, “Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems,” *IEEE Trans. Neural Networks*, vol. 5, pp. 594–603, July 1994.

[16] Q. Zhang and A. Benveniste, “Wavelet networks,” *IEEE Trans. Neural Networks*, vol. 3, pp. 889–898, Mar. 1992.

[17] Y. C. Pati and P. S. Krishnaprasad, “Analysis and synthesis of feedforward neural networks using discrete affine wavelet transforms,” *IEEE Trans. Neural Networks*, vol. 4, pp. 73–85, Jan. 1993.

[18] J. Zhang, G. G. Walter, Y. Miao, and W. G. W. Lee, “Wavelet neural networks for function learning,” *IEEE Trans. Signal Processing*, vol. 43, pp. 1485–1497, June 1995.

[19] Q. Zhang, “Using wavelet network in nonparametric estimation,” *IEEE Trans. Neural Networks,* vol. 8, pp. 227–236, Mar. 1997.

[20] Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks.* Reading, MA: Addison-Wesley, 1989.

[21] S.-S. Yang and C.-S. Tseng, “An orthonormal neural network for function approximation,” *IEEE Trans. Syst., Man, Cybern. B,* vol. 26, pp. 779–784, Oct. 1996.

[22] N. Sadegh, “A perceptron based neural network for identification and control of nonlinear systems,” *IEEE Trans. Neural Networks,* vol. 4, pp. 982–988, Nov. 1993.

[23] K. H. Chon and R. J. Cohen, “Linear and nonlinear ARMA model parameter estimation using an artificial neural network,” *IEEE Trans. Biomed. Eng.*, vol. 44, pp. 168–174, Mar. 1997.

[24] J. Teeter and M. Y. Chow, “Application of functional link neural network to HVAC thermal dynamic system identification,” *IEEE Trans. Ind. Electron.*, vol. 45, pp. 170–176, Feb. 1998.

[25] J. C. Patra, R. N. Pal, B. N. Chatterji, and G. Panda, “Identification of nonlinear dynamic systems using functional link artificial neural networks,” *IEEE Trans. Syst., Man, Cybern. B,* vol. 29, pp. 254–262, Apr. 1999.

[26] A. Namatame and N. Ueda, “Pattern classification with Chebyshev neural networks,” *Intl. J. Neural Networks,* vol. 3, pp. 23–31, Mar. 1992.

[27] T. T. Lee and J. T. Jeng, “The Chebyshev polynomial-based unified model neural networks for functional approximation,” *IEEE Trans. Syst., Man Cybern.-B,* vol. 28, pp. 925–935, Dec. 1998.

[28] M. F. Moller, “A scaled conjugate gradient algorithm for fast supervised learning,” *Neural Networks,* vol. 6, pp. 525–533, 1993.

[29] R. Battiti, “First- and second-order methods for learning: Between steepest descent and Newton’s method,” *Neural Comput.*, vol. 4, pp. 141–166, 1992.

[30] C. Charalambous, “Conjugate gradient algorithm for efficient training of artificial neural networks,” *Proc. Inst. Elect. Eng.*, vol. 139, pp. 301–310, Mar. 1992.

[31] M. T. Hagan and M. B. Menhaj, “Training feedforward networks with Marquardt algorithm,” *IEEE Trans. Neural Networks*, vol. 5, pp. 989–993, Nov. 1994.

[32] R. K. Madyastha and B. Aazhang, “An algorithm for training multilayer perceptrons for data classification and function interpolation,” *IEEE Trans. Circuits Syst.*, vol. 41, pp. 866–875, Dec. 1994.

[33] N. A. Gershenfeld and A. S. Weigend, “The future of time series: Learning and understanding,” in *Time Series Prediction: Forecasting the Future and Past*, A. S. Weigend and N. A. Gershenfeld, Eds. Reading, MA: Addison-Wesley, 1993, pp. 1–70.

[34] S. S. Yang and C. S. Tseng, “An orthonormal neural network for functional approximation,” *IEEE Trans. Syst., Man, Cybern. B,* vol. 26, pp. 779–785, Oct. 1996.

[35] J. C. Patra, R. N. Pal, R. Baliarsingh, and G. Panda, “Nonlinear channel equalization for QAM signal constellation using artificial neural networks,” *IEEE Trans. Syst., Man, Cybern. B,* vol. 29, pp. 262–271, Apr. 1999.

Biography

Jagdish C. Patra (M’96) received the B.S. and M.S. degrees in electronics and telecommunication engineering from Sambalpur University, India, in 1978 and 1989, respectively, and the Ph.D. degree in electronics and communication engineering from the Indian Institute of Technology, Kharagpur, in 1996.

After completion of the B.S. degree, he worked in different R&D, teaching, and Government organizations for about eight years. In 1987, he joined the Regional Engineering College, Rourkela, as a Lecturer, where he was promoted to an Assistant Professor in 1990. In April 1999, he went to the Technical University of Delft, Delft, the Netherlands, as a Guest Teacher (Gasdoscent) for six months. Subsequently, he joined the School of EEE, Nanyang Technological University (NTU), Singapore, as a Research Fellow in October 1999. Currently, he is serving as an Assistant Professor in the School of Computer Engineering, NTU. His research interest includes intelligent signal processing using neural networks, fuzzy sets, and genetic algorithms. He has published several research papers in international journals of repute and conferences.

Dr. Patra is a member of the Institution of Engineers (India).

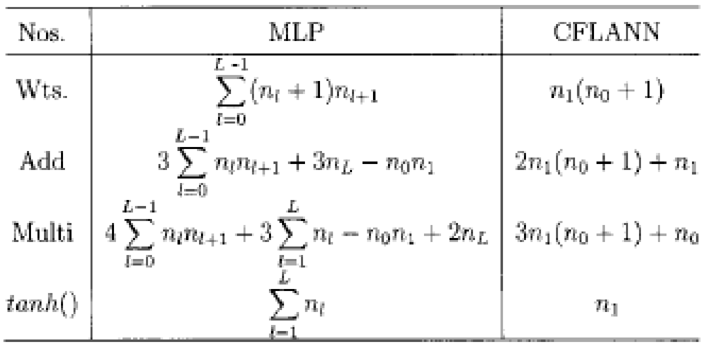
Alex C. Kot (SM’95) received the M.S. degree from the University of Rochester, NY and the Ph.D. degree in electrical engineering from the University of Rhode Island, RI, in 1989.

He was with AT&T Bell, New York. Since 1991, he has been with the Nanyang Technological University (NTU), Singapore, where he is currently Head of the Information Engineering Division. His research and teaching interests are in the areas of signal processing for communications, digital signal processing, biometrics, and information security.

Dr. Kot received the NTU Best Teacher of the Year Award. He is currently an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He has served as the General Co-Chair for the Second International Conference on Information, Communications and Signal Processing (ICICS) in December 1999. He is the Advisor for ICICS’01 and ICONIP’02. He has served as the Chairman of the IEEE Signal Processing Chapter in Singapore.

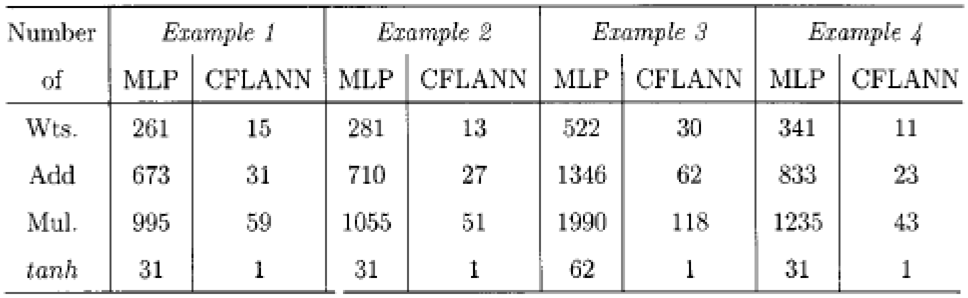
List of tables

List of figures



Nos.	MLP	CFLANN
Wts.	$\sum_{l=0}^{L-1}(n_l+1)n_{l+1}$	$n_1(n_0+1)$
Add	$3\sum_{l=0}^{L-1} n_l n_{l+1} + 3n_L - n_0 n_1$	$2n_1(n_0+1)+n_1$
Multi	$4\sum_{l=0}^{L-1} n_l n_{l+1} + 3\sum_{l=1}^{L} n_l - n_0 n_1 + 2n_L$	$3n_1(n_0+1)+n_0$
$tanh()$	$\sum_{l=1}^{L} n_l$	n_1

TABLE I



Number	*Example 1*		*Example 2*		*Example 3*		*Example 4*	
of	MLP	CFLANN	MLP	CFLANN	MLP	CFLANN	MLP	CFLANN
Wts.	261	15	281	13	522	30	341	11
Add	673	31	710	27	1346	62	833	23
Mul.	995	59	1055	51	1990	118	1235	43
tanh	31	1	31	1	62	1	31	1

TABLE II

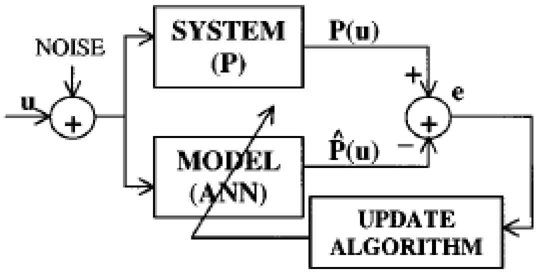


Fig. 1

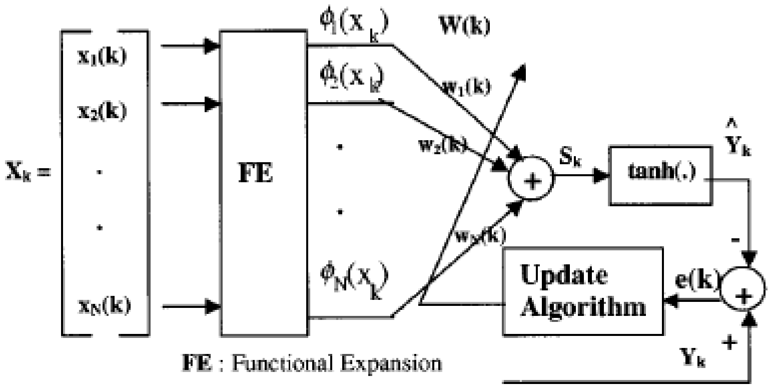

X_k =
$x_1(k)$
$x_2(k)$
$x_N(k)$
FE
$\phi_1(X_k)$
$\phi_2(X_k)$
$\phi_N(X_k)$
W(k)
$w_1(k)$
$w_2(k)$
$w_N(k)$
+
S_k
tanh(.)
$\hat{Y}_k$
-
e(k)
+
Update Algorithm
Y_k
+
FE : Functional Expansion

Fig. 2

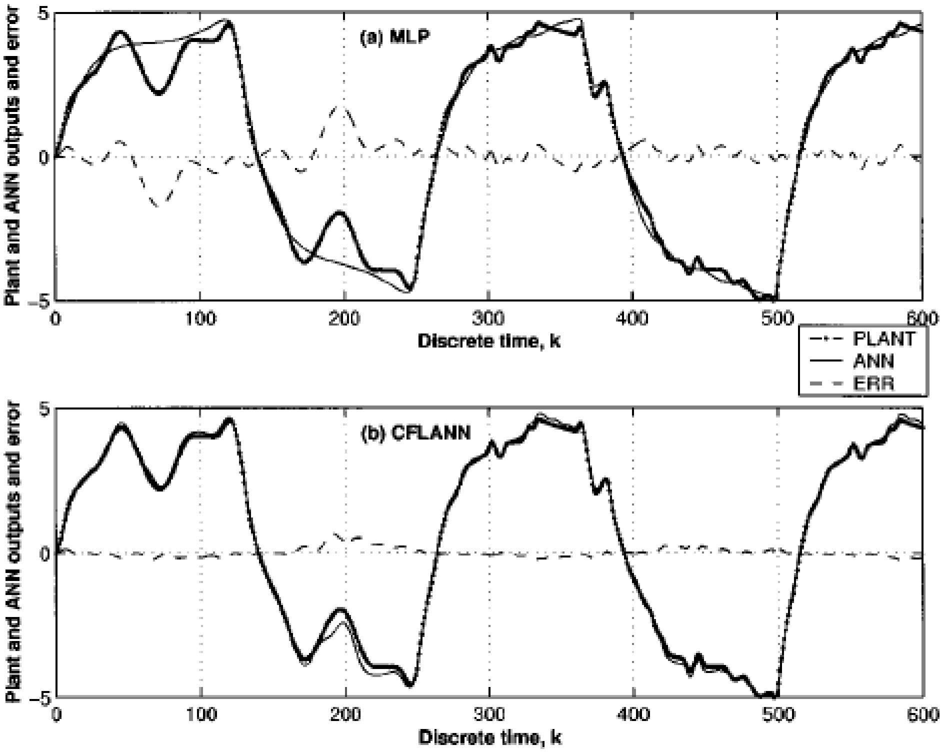

(a) MLP
Plant and ANN outputs and error
5
0
−5
0
100
200
300
400
500
600
Discrete time, k
PLANT
ANN
ERR
(b) CFLANN
Plant and ANN outputs and error
5
0
−5
0
100
200
300
400
500
600
Discrete time, k

Fig. 3

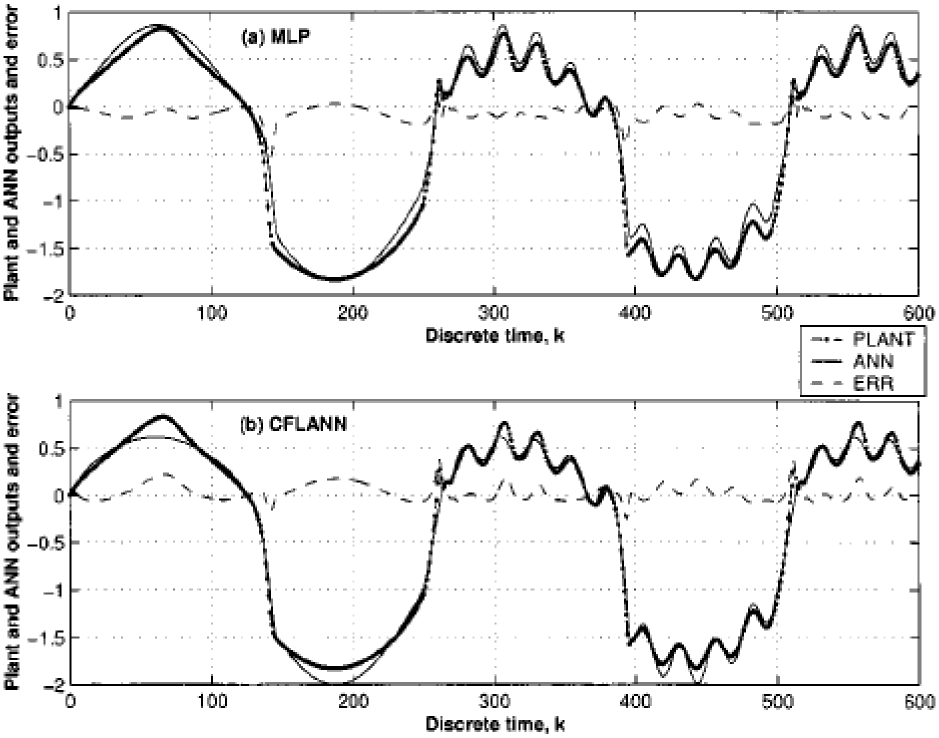

(a) MLP
Plant and ANN outputs and error
1
0.5
0
−0.5
−1
−1.5
−2
0
100
200
300
400
500
600
Discrete time, k
PLANT
ANN
ERR
(b) CFLANN
Plant and ANN outputs and error
1
0.5
0
−0.5
−1
−1.5
−2
0
100
200
300
400
500
600
Discrete time, k

Fig. 4

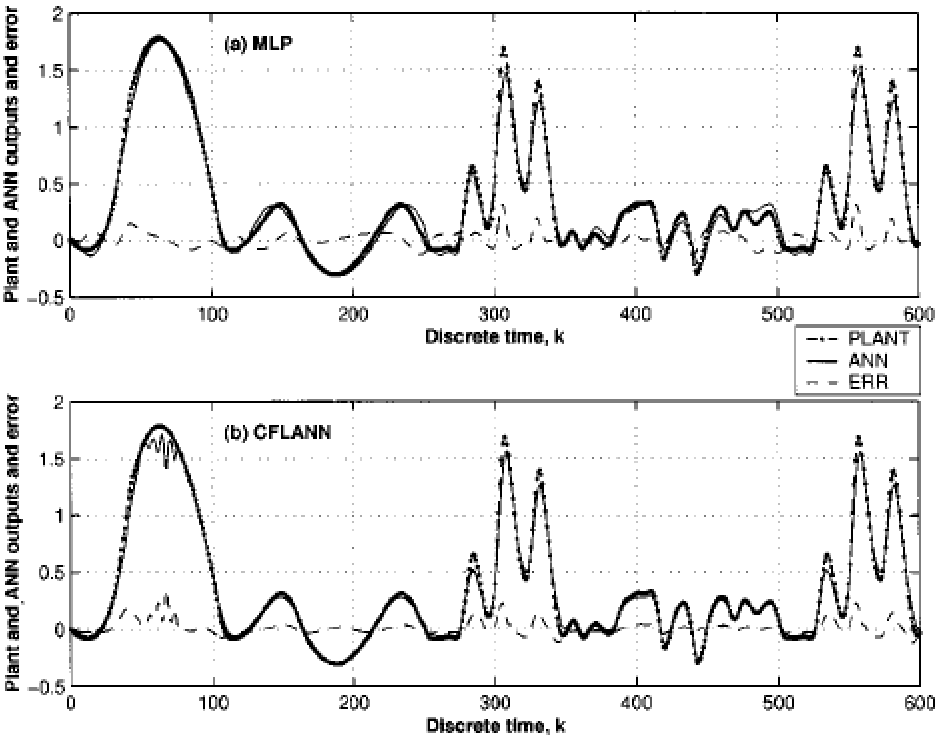

(a) MLP
Plant and ANN outputs and error
2
1.5
1
0.5
0
−0.5
0
100
200
300
400
500
600
Discrete time, k
PLANT
ANN
ERR
(b) CFLANN
Plant and ANN outputs and error
2
1.5
1
0.5
0
−0.5
0
100
200
300
400
500
600
Discrete time, k

Fig. 5

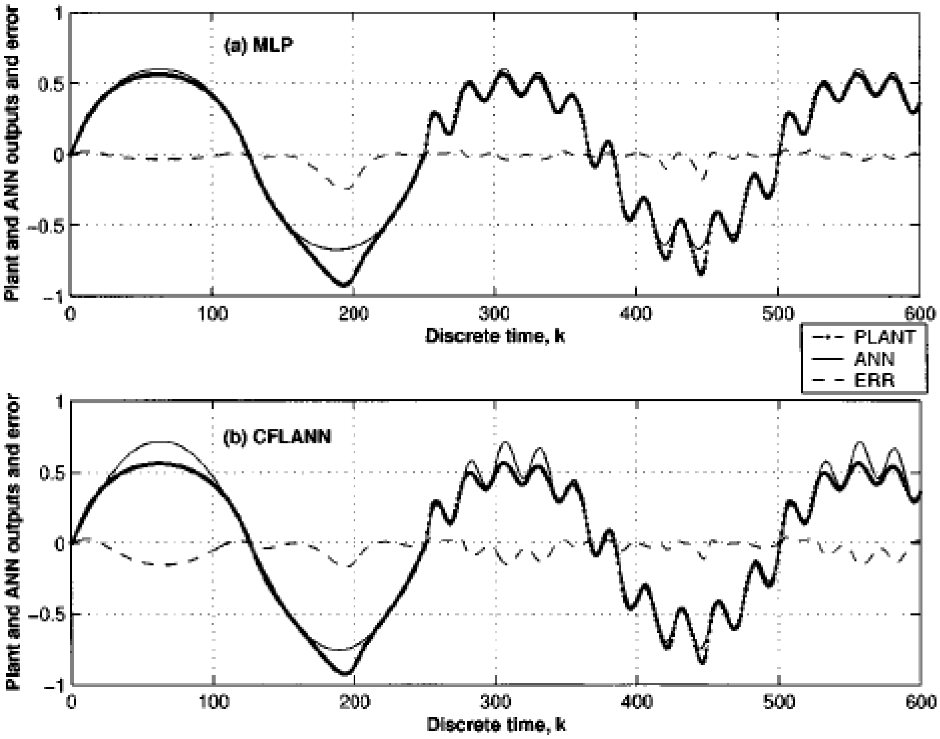

(a) MLP
Plant and ANN outputs and error
1
0.5
0
−0.5
−1
0
100
200
300
400
500
600
Discrete time, k
PLANT
ANN
ERR
(b) CFLANN
Plant and ANN outputs and error
1
0.5
0
−0.5
−1
0
100
200
300
400
500
600
Discrete time, k

Fig. 6