Vesna M. Ranković
Assistant Professor
University of Kragujevac
Faculty of Mechanical Engineering

Ilija Ž. Nikolić
Professor
University of Kragujevac
Faculty of Mechanical Engineering

Identification of Nonlinear Models with Feedforward Neural Network and Digital Recurrent Network

Nonlinear system identification via Feedforward Neural Networks (FNN) and Digital Recurrent Network (DRN) is studied in this paper. The standard backpropagation algorithm is used to train the FNN. A dynamic backpropagation algorithm is employed to adapt weights and biases of the DRN. The neural networks are trained using the identified error between the model's output and plant's output. Results of simulations show that the application of the FNN and DRN to identification of complex nonlinear dynamics gives satisfactory results.

Keywords: *neural network, nonlinear system, identification.*

1. INTRODUCTION

In general, dynamic systems are complex and nonlinear. An important step in nonlinear control is the development of a nonlinear model. In recent years, computational-intelligence techniques, such as neural networks, fuzzy logic and combined neuro-fuzzy systems algorithms have become very effective tools of identification and control of nonlinear plants. The problem of identification consists of choosing an identification model and adjusting the parameters such that the response of the model approximates the response of the real system to the same input.

Since 1986, neural network has been applied to the identification of nonlinear dynamical systems. Most of the works are based on multilayer feedforward neural networks with backpropagation learning algorithm. A novel multilayer discrete-time neural network is presented in [1] for identification of nonlinear dynamical systems.

In [2], a new scheme for on-line states and parameters estimation of a large class of nonlinear systems using RBF (Radial Basis Function) neural network has been designed.

A new approach to control nonlinear discrete dynamic systems, which relies on the identification of a discrete model of the system by a feedforward neural network with one hidden layer is presented in [3].

Nonlinear system identification via discrete-time recurrent single layer and multilayer neural networks are studied in [4]. An identification method for nonlinear models in the form of Fuzzy-Neural Networks is introduced in [5]. The Fuzzy-Neural Networks combines fuzzy "if-then" rules with neural networks.

The adaptive time delay neural network is used for identification of nonlinear systems in [6]. Four architectures are proposed for identifying different classes of nonlinear systems.

In [7] is investigated the identification of nonlinear systems by feedforward neural networks, radial basis function neural networks, Runge-Kutta neural networks and adaptive neuro-fuzzy inference systems. The result of simulation, reported in this paper, indicates that adaptive neuro fuzzy inference systems are a good candidate for identification purposes.

This paper investigates the identification of nonlinear system by FNN and DRN. In Section 2 the nonlinear system identification is analyzed. The structures of the FNN and RNN are presented in Section 3. Some simulation results and discussions related to system identification are provided in Section 4. Section 5 gives the concluding remarks.

Received: March 2008, Accepted: August 2008
Correspondence to: Dr Vesna Ranković
Faculty of Mechanical Engineering,
Sestre Janjić 6, 34000 Kragujevac, Serbia
E-mail: vesnar@kg.ac.yu

2. METHODS FOR NONLINEAR SYSTEMS IDENTIFICATION

Different methods have been developed in the literature for nonlinear system identification. These methods use a parameterized model. The parameters are updated to minimize an output identification error.

A wide class of nonlinear dynamic systems with an input u and an output y can be described by the model:

$$y_m(k) = f_m(\varphi(k), \theta) \tag{1}$$

where $y_m(k)$ is the output of the model, $\varphi(k)$ is the regression vector and θ is the parameter vector.

Depending on the choice of the regressors in $\varphi(k)$, different models can be derived:

NFIR (Nonlinear Finite Impulse Response) model:

$$\varphi(k) = \left(u(k-1), u(k-2), ..., u(k-n_u)\right),$$

where n_u denotes the maximum lag of the input.

NARX (Nonlinear AutoRegressive with eXogenous inputs) model:

$$\varphi(k) = \left(u(k-1), u(k-2), ..., u(k-n_u),\right.$$
$$\left. y(k-1), y(k-2), ..., y(k-n_y)\right),$$

where n_y denotes the maximum lag of the output.

NARMAX (Nonlinear AutoRegressive Moving Average with eXogenous inputs) model:

$$\varphi(k)=\left(u(k-1),u(k-2),...,u(k-n_u),y(k-1),\right.$$
$$\left.y(k-2),...,y(k-n_y),e(k-1),e(k-2),...,e(k-n_e)\right),$$

where e (k) is the prediction error and n_e is the maximum lag of the error.

NOE (Nonlinear Output Error) model:

$$\varphi(k)=\left(u(k-1),u(k-2),...,u(k-n_u),\right.$$
$$\left.y_m(k-1),y_m(k-2),...,y_m(k-n_y)\right).$$

NBJ (Nonlinear Box-Jenkins) model: uses all four regressor types.

The NARX and NOE models (Figs 1 and 2) are the most important representations of nonlinear systems.

3. FNN AND DRN NEURAL NETWORK FOR NONLINEAR SYSTEM IDENTIFICATION

Neural networks can be classified as feedforward and recurrent (or feedback). The two-layer feedforward neural network with sigmoidal activation function in the hidden layer and linear activation function in output layer has the ability to approximate nonlinear function if the number of neurons in the hidden layer is sufficiently large. The FNN used in this paper is shown in Figure 3.

The input vector to the neural network is defined as:

$$I^T(k)=\left[u(k-1),\,u(k-2)...u(k-n_u),y(k-1),\right.$$
$$\left.y(k-2)...y(k-n_y)\right].$$

The inputs u (k – 1), u (k – 2),…, u (k – n_u) and u (k – 1), u (k – 2),…, u (k – n_y) are multiplied by weights $\omega_{u_{ij}}$ and $\omega_{y_{ij}}$, respectively, and summed at each hidden node. Then the summed signal at a node activates a nonlinear function (sigmoid function). Thus, the output y (k) at a linear output node can be calculated from its inputs as follows:

$$y_m(k)=\sum_{i=1}^{n_H}\omega_i\frac{1}{1+e^{-\left(\sum_{j=1}^{n_u}u(k-j)\omega_{u_{ij}}+\sum_{j=1}^{n_y}y(k-j)\omega_{y_{ij}}+b_i\right)}}+b \quad (2)$$

where $n_u + n_y$ is the number of inputs, n_H is the number of hidden neurons, $\omega_{u_{ij}}$ is the first layer weight between the input u (k – j) and the i-th hidden neuron, $\omega_{y_{ij}}$ is the first layer weight between the input y (k – j) and the i-th hidden neuron, ω_i is the second layer weight between the i-th hidden neuron and output neuron, b_i is a biased weight for the i-th hidden neuron and b is a biased weight for the output neuron.

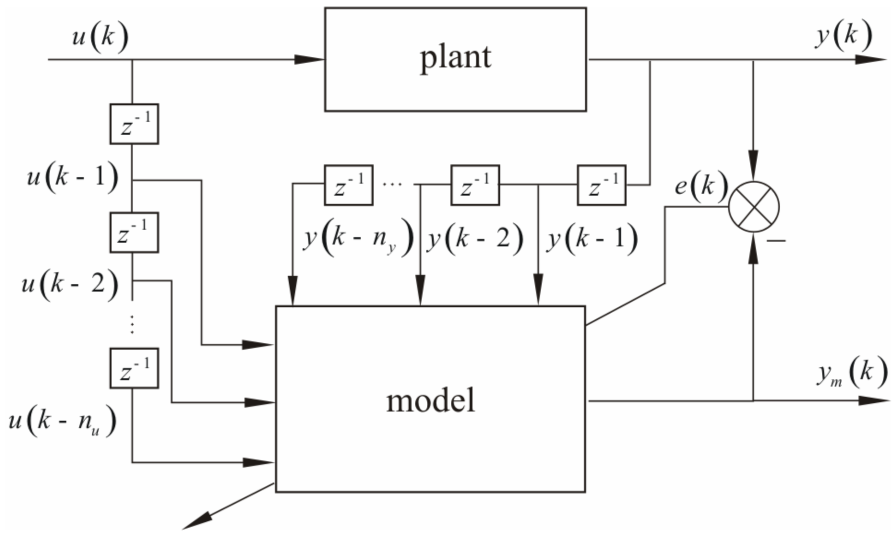

Figure 1. The general block scheme of the NARX model [8]

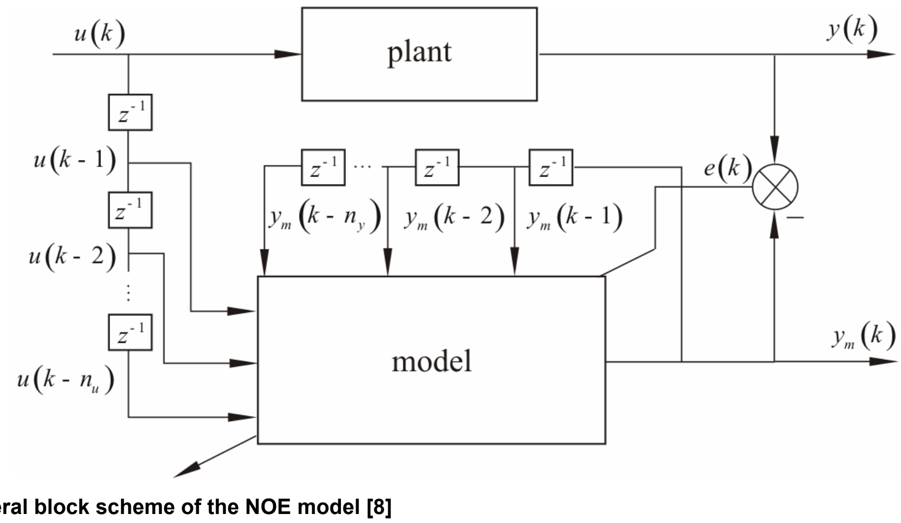

Figure 2. The general block scheme of the NOE model [8]

It can be seen from Figure 3 that the FNN is a realization of the NARX model (Fig. 1). The difference between the output of the plant y (k) and the output of the network y_m (k) is called the prediction error:

$$e(k) = y(k) - y_m(k) \tag{3}$$

This error is used to adjust the weights and biases in the network via the minimization of the following function:

$$\varepsilon = \frac{1}{2}\left[y(k) - y_m(k)\right]^2 \tag{4}$$

The backpropagation is the most popular algorithm to train FNN [9].

The backpropagation update rule for the weights ($\omega_{u_{ij}}$, $\omega_{y_{ij}}$) and bisaes (b_i, b) is:

$$\omega_{u_{ij}}(k+1) = \omega_{u_{ij}}(k) - \eta\frac{\partial\varepsilon}{\partial\omega_{u_{ij}}} \tag{5}$$

$$\omega_{y_{ij}}(k+1) = \omega_{y_{ij}}(k) - \eta\frac{\partial\varepsilon}{\partial\omega_{y_{ij}}} \tag{6}$$

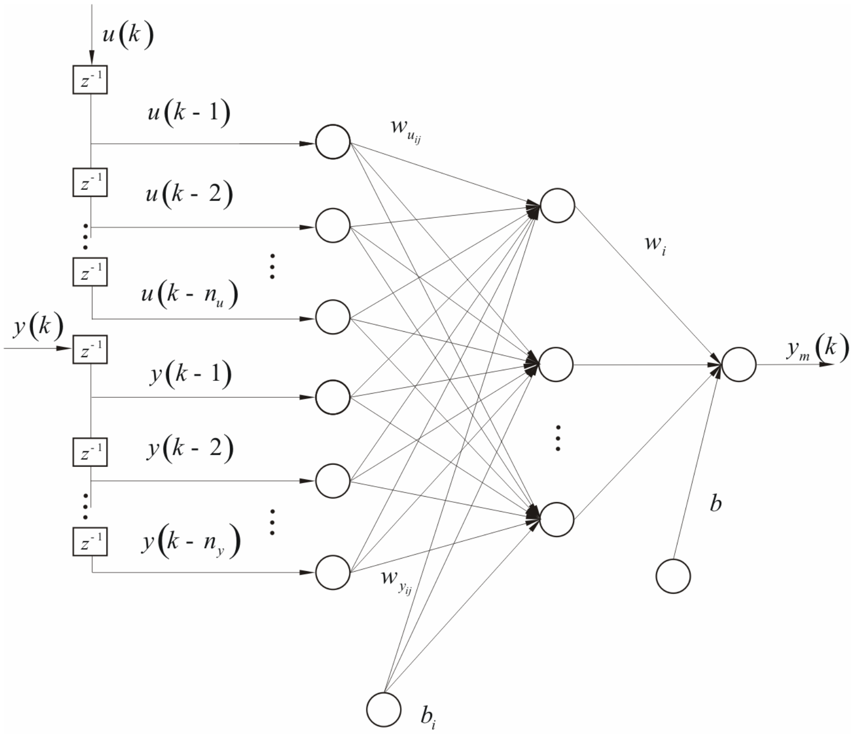

Figure 3. Feedforward neural network structure [3]

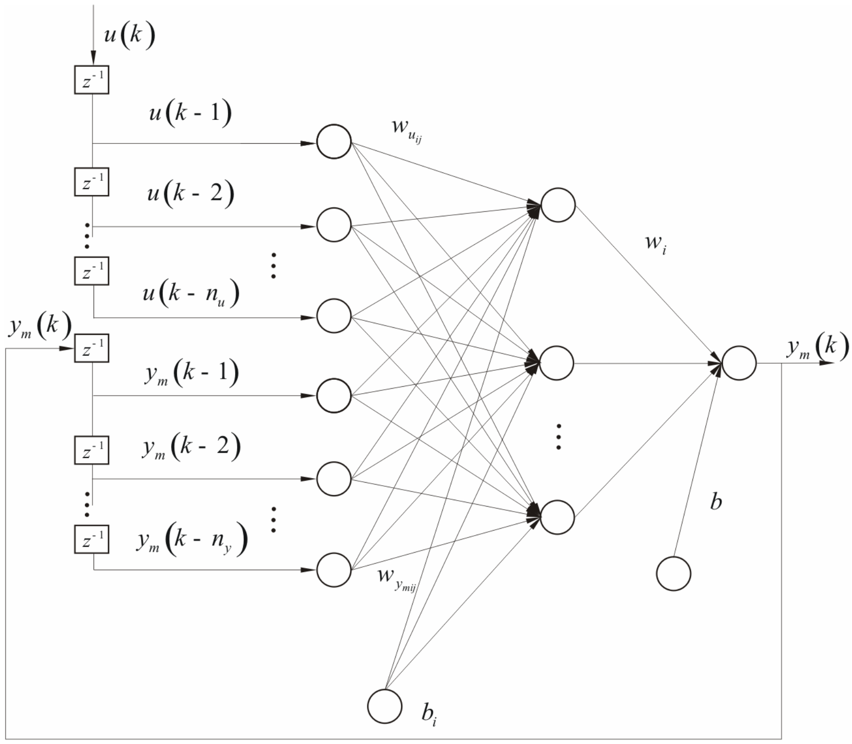

Figure 4. Digital Recurrent Network [9]

$$b_i(k+1) = b_i(k) - \eta \frac{\partial \varepsilon}{\partial b_i} \quad (7)$$

$$b(k+1) = b(k) - \eta \frac{\partial \varepsilon}{\partial b} \quad (8)$$

where η is the update rate and:

$$\frac{\partial \varepsilon}{\partial \omega_{u_{ij}}} = \frac{\partial \varepsilon}{\partial y_m} \frac{\partial y_m}{\partial \omega_{u_{ij}}}; \quad \frac{\partial \varepsilon}{\partial \omega_{y_{ij}}} = \frac{\partial \varepsilon}{\partial y_m} \frac{\partial y_m}{\partial \omega_{y_{ij}}};$$

$$\frac{\partial \varepsilon}{\partial b_i} = \frac{\partial \varepsilon}{\partial y_m} \frac{\partial y_m}{\partial b_i}; \quad \frac{\partial \varepsilon}{\partial b} = \frac{\partial \varepsilon}{\partial y_m} \frac{\partial y_m}{\partial b}$$

Figure 4 is an example of a DRN. The output of the network is feedback to its input. This is a realization of the NOE model, Fig. 2. The output of the network is a function not only of the weights, biases, and network input, but also of the outputs of the network at previous points in time. In [10] dynamic backpropagation algorithm is used to adapt weights and biases.

DRN network is composed of a nonlinear hidden layer and a linear output layer. The inputs u (k – 1), u (k – 2),…, u (k – n_u) are multiplied by weights $\omega_{u_{ij}}$, outputs y_m (k – 1), y_m (k – 2),…, y_m (k – n_y) are multiplied by weights $\omega_{y_{ij}}$ and summed at each hidden node. Then the summed signal at a node activates a nonlinear function. The hidden neurons activation function is the hyperbolic tangent sigmoid function. In Figure 4, ω_i represents the weight that connects the node i in the hidden layer and the output node; b_i represents the biased weight for i-th hidden neuron and b is a biased weight for the output neuron.

The output of the network is:

$$y_m(k) = \sum_{i=1}^{n_H} \omega_i \nu_i + b \quad (9)$$

where n_H is the number of hidden nodes and:

$$\nu_i = \frac{e^{n_i} - e^{-n_i}}{e^{n_i} + e^{-n_i}} \quad (10)$$

$$n_i = \sum_{j=1}^{n_u} u(k-j)\omega_{u_{ij}} + \sum_{j=1}^{n_y} y_m(k-j)\omega_{y_{mij}} + b_i \quad (11)$$

The network should learn the $\omega_{u_{ij}}$, $\omega_{y_{ij}}$, ω_i, b_i and b that minimizes ε (4).

Using the gradient decent, the weight and bias updating rules can be described as:

$$\omega_{u_{ij}}(k+1) = \omega_{u_{ij}}(k) - \eta \frac{\partial \varepsilon}{\partial \omega_{u_{ij}}} \quad (12)$$

$$\omega_{y_{mij}}(k+1) = \omega_{y_{mij}}(k) - \eta \frac{\partial \varepsilon}{\partial \omega_{y_{mij}}} \quad (13)$$

$$b_i(k+1) = b_i(k) - \eta \frac{\partial \varepsilon}{\partial b_i} \quad (14)$$

$$b(k+1) = b(k) - \eta \frac{\partial \varepsilon}{\partial b} \quad (15)$$

where:

$$\frac{\partial \varepsilon}{\partial \omega_{u_{ij}}} = \frac{\partial^e \varepsilon}{\partial y_m} \frac{\partial y_m}{\partial \omega_{u_{ij}}}; \quad \frac{\partial \varepsilon}{\partial \omega_{y_{ij}}} = \frac{\partial^e \varepsilon}{\partial y_m} \frac{\partial y_m}{\partial \omega_{y_{ij}}};$$

$$\frac{\partial \varepsilon}{\partial b_i} = \frac{\partial^e \varepsilon}{\partial y_m} \frac{\partial y_m}{\partial b_i}; \quad \frac{\partial \varepsilon}{\partial b} = \frac{\partial^e \varepsilon}{\partial y_m} \frac{\partial y_m}{\partial b}$$

where the superscript e indicates an explicit derivative, not accounting for indirect effects through time.

The terms $\frac{\partial y_m}{\partial \omega_{u_{ij}}}$, $\frac{\partial y_m}{\partial \omega_{y_{ij}}}$, $\frac{\partial y_m}{\partial b_i}$ and $\frac{\partial y_m}{\partial b}$ must be propagated forward through time, [10].

4. SIMULATION RESULTS

Example: We consider the dynamic system which is described by the following nonlinear difference equation [11]:

$$y(k) = 0.35 \frac{y(k-1)y(k-2)\left[y(k-1)+2.5\right]}{1+y^2(k-1)+y^2(k-2)} + 0.35u(k-1) \quad (16)$$

where y is the output of the plant and u is the plant input.

We assume that structure of the model is known, n_u = 1, n_y = 2. The inputs and output of the FNN and DRN are u (k – 1), y (k – 1), y (k – 2) and y (k), respectively.

The input u (k) was assumed to be a random signal uniformly distributed in the interval [-1 1]. In Figure 5 there are plotted the input signal applied to plant (16) and the corresponding response. The dashed line denotes the input. In this simulation the plant output is bounded within the approximation region [-0.4 0.6]. The training data for this example consists of 6000 observations.

The FNN is chosen to be a two-layer structure (Fig. 3) with 12 hidden neurons and a learning rate 0.85. The number of hidden neurons of the DRN structure is 12.

The learning rate was taken as 0.75. In these simulations different numbers of hidden neurons are tested. Choosing the right number of hidden neurons is essential for a good result. The total number of the parameters of each neural network is 49. In the learning processes, the weights (36) of the neural networks were adapted as well as the biases (13).

To validate the models, the input signal:

$$\begin{aligned} &u(t) = \sin(\pi t/25),\ t < 250 \\ &u(t) = 0.3\sin(\pi t/25) + 0.4\sin(\pi t/32) + \\ &\qquad 0.3\sin(\pi t/40),\ 250 \le t \le 1000 \end{aligned} \quad (17)$$

was used.

Figure 6 illustrates prediction error. From the Figure 6 it can be seen that maximum prediction error is less than 0.015.

The output of the DRN follows of the output of the plant. The simulation result indicates that the prediction error is less than $0.7 \cdot 10^{-3}$, Fig. 7.

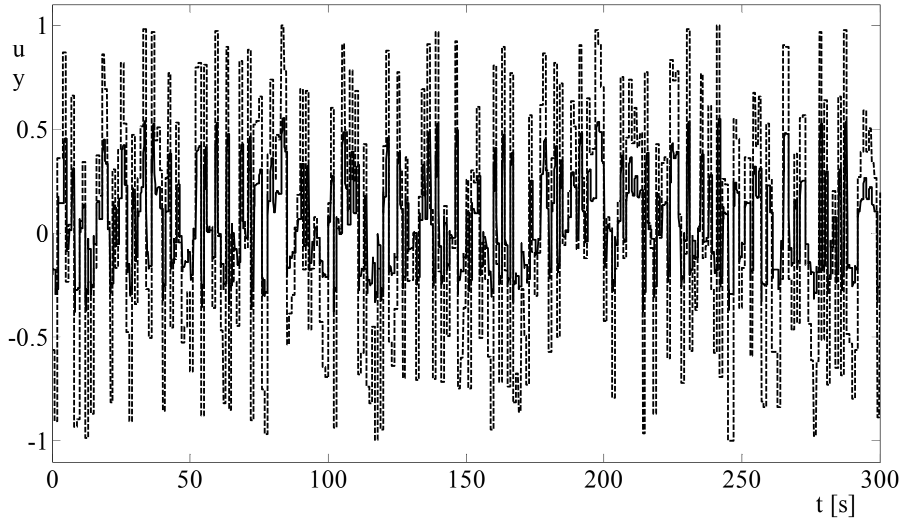

Figure 5. Identification data (input-dashed line, output-solid line)

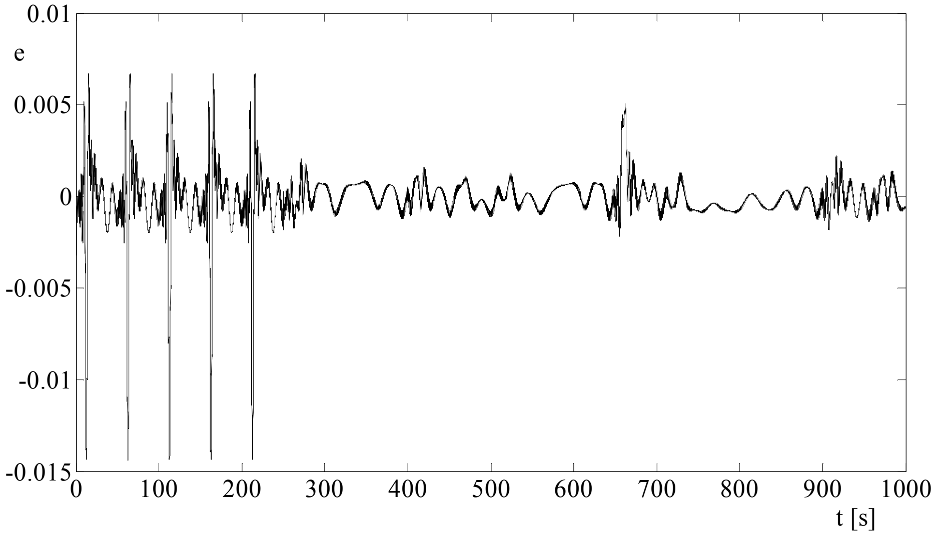

Figure 6. Prediction error (FNN model)

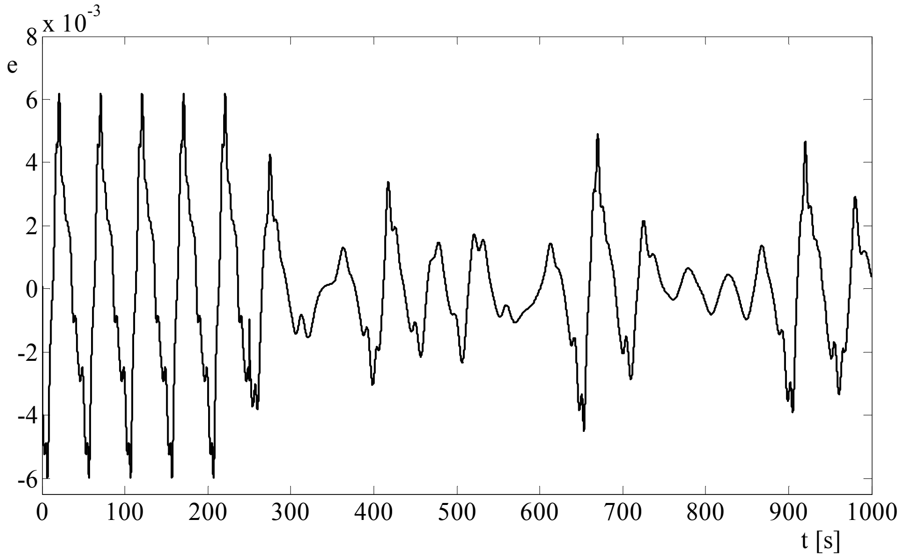

Figure 7. Prediction error (DRN model)

5. CONCLUSION

The dynamical systems contain nonlinear relations which are difficult to model with conventional techniques. In this paper, the FNN and DRN have been successfully applied to unknown nonlinear system identification and modeling. In the designing of neural network model, the problem is how to determine an optimal architecture of network.

The determination of the values of n_u and n_y is an open question. Large time lags result in better prediction of the NN. However, large n_u and n_y also result in a large number of parameters (weights and biases) that need to be adapted.

Fuzzy logic and neuro-fuzzy systems (ANFIS) have been applied to identification of nonlinear dynamics.

However, neural networks are the simplest approaches in the sense of computational complexity, [12].

FNN and DRN models can be embedded into a model predictive control scheme. If the plant output can not be measured, DRN model can be taken.

The results obtained can be extended to multivariable systems. Industrial robots have to face many uncertainties in their dynamics, in particular structured uncertainty, such as payload parameter, and unstructured one, such as friction and disturbance. The FNN and DRN have been successfully applied to identification of robot's dynamics.

ACKNOWLEDGMENT

This research was supported by the Ministry of Science and Environmental Protection of Republic of Serbia through the Mathematical Institute SANU Belgrade, Grants No. 144002.

REFERENCES

[1] Jagannathan, S. and Lewis, F.L.: Identification of nonlinear dynamical systems using multilayered neural networks, Automatica, Vol. 32, No. 12, pp. 1707-1712, 1996.

[2] Kenné, G., Ahmed-Ali, T., Lamnabhi-Lagarrigue, F. and Nkwawo, H.: Nonlinear systems parameters estimation using radial basis function network, Control Engineering Practice, Vol. 14, No. 7, pp. 819-832, 2006.

[3] Canelon, J.I., Shieh, L. and Karayiannis, N.B.: A new approach for neural control of nonlinear discrete dynamic systems, Information Sciences, Vol. 174, No. 3-4, pp. 177-196, 2005.

[4] Yu, W.: Nonlinear system identification using discrete-time recurrent neural networks with stable learning algorithms, Information Sciences, Vol. 158, pp. 131-147, 2004.

[5] Oh, S.-K., Pedrycz, W. and Park, H.-S.: Hybrid identification in fuzzy-neural networks, Fuzzy Sets and Systems, Vol. 138, No. 2, pp. 399-426, 2003.

[6] Yazdizadeh, A. and Khorasani, K.: Adaptive time delay neural network structures for nonlinear system identification, Neurocomputing, Vol. 47, No. 1-4, pp. 207-240, 2002.

[7] Onder Efe, M. and Kaynak, O.: A comparative study of neural network structures in identification of nonlinear systems, Mechatronics, Vol. 9, No. 3, pp. 287-300, 1999.

[8] Van Gorp, J.,: *Nonlinear Identification With Neural Networks and Fuzzy Logic*, PhD thesis, Vrije Universiteit Brussel, Brussel, 2000.

[9] Rumelhart, D.E., Hinton, G.E. and Williams, R.J.: Learning representations by back-propagating errors, Nature, Vol. 323, No. 6088, pp. 533-536, 1986.

[10] Hagan, M.T., de Jesús, O. and Schultz, R.: Training Recurrent Neural Networks for Filtering and Control, in: Medsker, L.R. and Jain, L.C. (Eds.): *Recurrent Neural Networks: Design and Applications*, CRC Press, Boca Raton, pp. 325-354, 1999.

[11] Sastry, P.S., Santharam, G. and Unnikrishnan, K.P.: Memory neuron networks for identification and control of dynamical systems, IEEE Transactions on Neural Networks, Vol. 5, No. 2, pp. 306-319, 1994.

[12] Onder Efe, M. and Kaynak, O.: A comparative study of soft-computing methodologies in identification of robotic manipulators, Robotics and Autonomous Systems, Vol. 30, No. 3, pp. 221-230, 2000.

NOMENCLATURE

$y_m(k)$	output of the model
$y(k)$	output of the system
$u(k)$	system input
$\boldsymbol{\varphi}(k)$	regression vector
$\boldsymbol{\theta}$	parameter vector
n_u	maximum lag of the input
n_y	maximum lag of the output
$e(k)$	prediction error
n_e	maximum lag of the error
n_H	number of hidden neurons
$\omega_{u_{ij}}$	first layer weight between the input $u(k-j)$ and the i-th hidden neuron
$\omega_{y_{ij}}$	first layer weight between the input $y(k-j)$ and the i-th hidden neuron
ω_i	second layer weight between the i-th hidden neuron and output neuron
b_i	biased weight for the i-th hidden neuron
b	biased weight for the output neuron

Notation

FNN	Feedforward Neural Networks
DRN	Digital Recurrent Network

ИДЕНТИФИКАЦИЈА НЕЛИНЕАРНИХ МОДЕЛА НЕУРОНСКОМ МРЕЖОМ СА ПРОСТИРАЊЕМ СИГНАЛА УНАПРЕД И ДИГИТАЛНОМ РЕКУРЕНТНОМ НЕУРОНСКОМ МРЕЖОМ

Весна М. Ранковић, Илија Ж. Николић

У овом раду се проучава идентификација нелинеарног система неуронском мрежом са простирањем сигнала унапред (*FNN*) и дигиталном рекурентном неуронском мрежом (*DRN*). За обучавање *FNN* користи се стандардни алгоритам са пропагацијом грешке уназад. Динамички алгоритам са пропагацијом грешке уназад употребљава се за адаптацију тежинских коефицијената и прагова активације *DRN*. Неуронске мреже се обучавају коришћењем идентификоване грешке између излаза модела и излаза објекта. Резултати симулација показују да примена *FNN* и *DRN* у идентификацији сложене нелинеарне динамике система даје задовољавајуће резултате.