

Neural Network Model Predictive Control of Nonlinear Systems Using Genetic Algorithms

V. Ranković, J. Radulović, N. Grujović, D. Divac

Vesna Ranković, Jasna Radulović, Nenad Grujović

Faculty of Mechanical Engineering, University of Kragujevac
Department for Applied Mechanics and Automatic Control
Serbia, 34000 Kragujevac; Sestre Janjić 6
E-mail: vesnar@kg.ac.rs, jasna@kg.ac.rs, gruja@kg.ac.rs

Dejan Divac

Institute for Development of Water Resources "Jaroslav Černi"
Serbia, 11000 Belgrade
Jaroslava Černog St., 11226 Beli Potok
E-mail: ddivac@eunet.rs

Abstract: In this paper the synthesis of the predictive controller for control of the nonlinear object is considered. It is supposed that the object model is not known. The method is based on a digital recurrent network (DRN) model of the system to be controlled, which is used for predicting the future behavior of the output variables. The cost function which minimizes the difference between the future object outputs and the desired values of the outputs is formulated. The function *ga* of the Matlab's Genetic Algorithm Optimization Toolbox is used for obtaining the optimum values of the control signals. Controller synthesis is illustrated for plants often referred to in the literature. Results of simulations show effectiveness of the proposed control system.

Keywords: model predictive control, nonlinear system, identification, digital recurrent network, genetic algorithm.

1 Introduction

The predictive controllers are based on the mathematical model of the object, which is being controlled. Nonlinear system identification and prediction is a complex task. All the processes in nature are nonlinear. In large number of processes, the nonlinearities are not prominent, so their behavior can be described by the linear model. In the linear systems theory there exist a large number of methods that can be applied for obtaining the linear model of processes. The nonlinear model must be chosen when the nonlinearity is strongly exhibited. In the identification process, the parameters of the mathematical model are being determined as such that the difference between the system response and its mathematical model is as least as possible, both in the transient regime and in stationary state. The general model of linear processes is ARX (Auto Regressive eXogenous), while for the nonlinear ones it is NARX (Nonlinear Auto Regressive eXogenous). The NARX model structure enables application of the neural networks, the fuzzy systems and the neuro-fuzzy systems for approximation of the nonlinear function.

Neural networks have been applied to the identification of nonlinear dynamical systems. The most of the works are based on multilayer feedforward neural networks with backpropagation learning algorithm. However, the conventional back-propagation algorithm has the problems of local minima and slow rate of convergence. A novel multilayer discrete-time neural network is presented for the identification of nonlinear dynamical systems, [1]. In [2] a new scheme for on-line states and parameters estimation of a large class of nonlinear systems using 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]. In [5] an identification method for nonlinear models in the form of fuzzy-neural networks is introduced. The fuzzy-neural networks combine fuzzy if-then rules with neural networks. The adaptive time delay neural network is used for the identification of nonlinear systems [6], and four architectures are proposed for identifying different classes of nonlinear systems. The identification of nonlinear systems by feedforward neural networks, radial basis function neural networks, Runge-Kutta neural networks and adaptive neuro-fuzzy inference systems is investigated in [7]. Result of simulation indicates that adaptive neuro fuzzy inference systems are a good candidate for identification purposes. However, neural networks are the simplest approaches in the sense of computational complexity. In [8] nonlinear system identification via feedforward neural network and digital recurrent network is studied.

Model predictive control (MPC) is applied to a large number of nonlinear industrial process, [9,10,11,12]. The methodology to design and implement neural predictive controllers for nonlinear system has been developed in [13,14,15].

In [13] feedforward neural networks to estimate the nonlinear process are applied. Also, for the minimization of the cost function, the Matlab's Optimal Toolbox functions *fminunc* and *fmincon* were used. The design methodology for predictive control of industrial processes via recurrent fuzzy neural networks is presented in [14]. In [15] the multilayer perceptron is used to identification of the nonlinear object and genetic algorithm is applied to solve the multi-criteria optimization problem.

In this paper the control of the nonlinear object is studied and it is identified by the DRN. Recurrent networks are more powerful than nonrecurrent networks and have important uses in control and signal processing applications, [16]. They have been shown to be more efficient than feedforward neural networks in terms of the number of neurons required to model a dynamic system [17,18]. Models with recurrent networks are shown to have the capability of capturing various plant nonlinearities, [19,20].

The major objective of the study presented in this paper is to take advantages of the recurrent neural networks for modelling and genetic algorithms for optimization. The proposed method formulates a dynamic nonlinear optimization problem, where the cost function consists of two terms: the differences between the DRN model predictions and the desired output trajectory over a prediction horizon and the control energy over a control horizon. For the solution of nonlinear optimization problem, a genetic algorithm is used, which can approximate the optimum solution very fast, compared to conventional optimization techniques. In this paper, the genetic algorithm has been successfully used in combinations with digital recurrent network.

In the second section the identification of the nonlinear object by application of the DRN is explained. In section three the principle of work of the predictive controllers is analyzed. In this paper for obtaining the optimum values of the control signals, the genetic algorithm (GA) was used. Results of simulations are given in section four, while section five presents the concluding remarks.

2 Neural network for identification of nonlinear dynamic

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 and an output can be described by the model:

$$y_m(k) = f_m(\varphi(k), \theta), \quad (1)$$

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 -

$$\varphi(k) = (u(k-1), u(k-2), \dots, u(k-n_u)),$$

where n_u denotes the maximum lag of the input.

- **NARX** (Nonlinear AutoRegressive with eXogenous inputs) model -

$$\varphi(k) = (u(k-1), u(k-2), \dots, u(k-n_u), y(k-1), y(k-2), \dots, y(k-n_y)),$$

where n_y denotes the maximum lag of the output.

- **NARMAX** (Nonlinear AutoRegressive Moving Average with eXogenous inputs) model -

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

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

- **NOE** (Nonlinear Output Error) model -

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

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

The NARX and NOE are the most important representations of nonlinear systems. The block scheme of the DRN model which corresponded to the NOE model is shown in Figure 1.

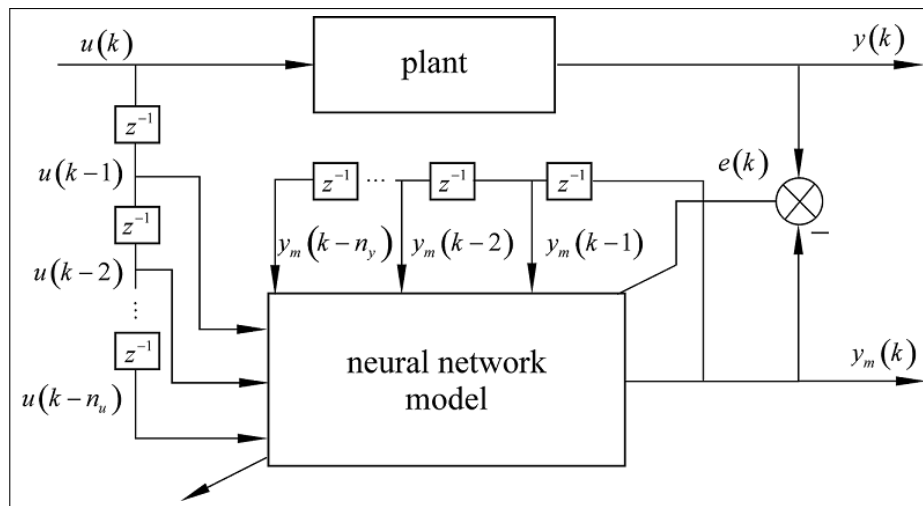


Figure 1: The block scheme of the neural network model

Figure 2 is an example of a DRN. The output of the network is feedback to its input. 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 [16] 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 $u(k-1), u(k-2), \dots, u(k-n_u)$ are multiplied by weights ω_{yij} outputs $y_m(k-1), y_m(k-2), \dots, y_m(k-n_y)$ are

multiplied by weights ω_{yij} 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 2, 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 is a biased weight for the output neuron.

The output of the network is:

$$y_m(k) = \sum_{i=1}^{n_H} \omega_i v_i + b \quad (2)$$

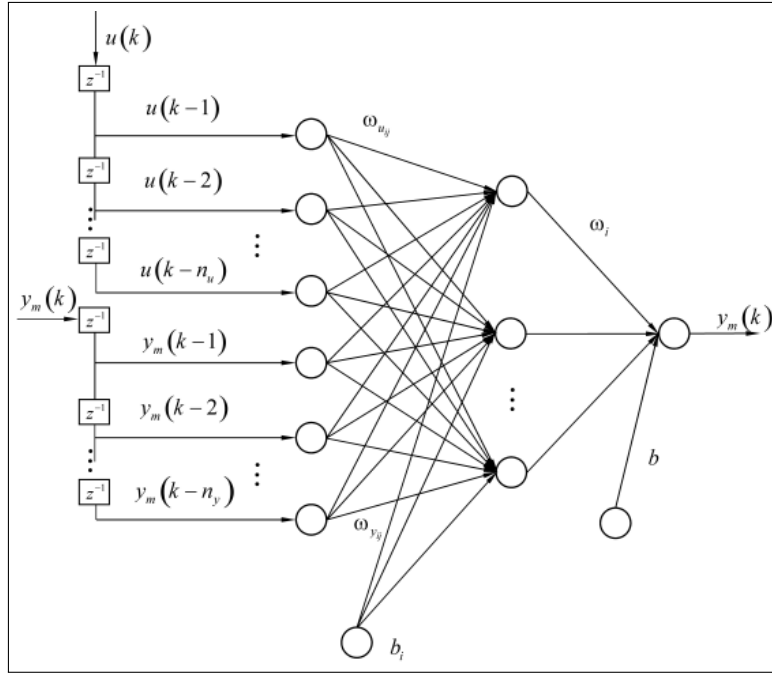


Figure 2: Digital Recurrent Network

where n_H is the number of hidden nodes and:

$$v_i = \frac{e^{n_i} - e^{-n_i}}{e^{n_i} + e^{-n_i}} \quad (3)$$

$$n_i = \sum_{j=1}^{n_u} u(k-j)\omega_{u_{ij}} + \sum_{j=1}^{n_y} y_m(k-j)\omega_{y_{ij}} + b_i \quad (4)$$

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

$$\varepsilon = \frac{1}{2} [y(k) - y_m(k)]^2 \quad (5)$$

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

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

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

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

$$b(k+1) = b(k) - \eta \frac{\partial \varepsilon}{\partial b} \quad (9)$$

where:

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

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, [16].

3 Model predictive control

Here the principle of operation of the predictive controllers will be briefly presented. Let us suppose that the mathematical model of the process is known. Based on the model, it is possible to determine the future outputs from the object $y(k+1)$, $l = 1, 2, \dots, N_H$, where N_H is the prediction horizon. The future outputs depend on the object current states and future control signals, $u(k+l)$, $l = 1, 2, \dots, N_c$, where N_c is the control horizon and $N_c \leq N_H$. The predictive controllers compute potential future control signals such that the future outputs will be as close as possible to the desired values $r(k+l)$, $l = 1, 2, \dots, N_H$. Figure 3 shows the basic concept of the model predictive control.

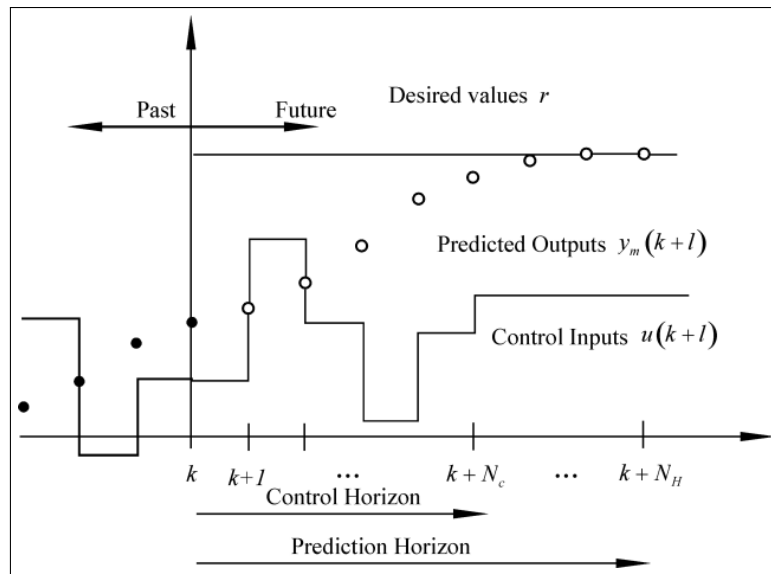


Figure 3: Basic concept of a model predictive control method

There are several types of the predictive controllers. In this work the GPC (General Predictive Control) controller is used, where the cost function is calculated as:

$$J(k) = \sum_{l=1}^{N_H} \left[r(k+l) - y_m(k+l) \right]^2 + \alpha \sum_{l=1}^{N_c} \Delta u^2(k+l-1), \quad (10)$$

where:

$$\Delta u^2(k+l-1) = u(k+l-1) - u(k+l-2),$$

and α is the weight factor of the control signal.

Real processes are subject to constraints. We consider constraints which limit the range of the control signal, the gradient of the control signal and the future model predictions:

$$\begin{aligned} u_{min} &\leq u(k+l) \leq u_{max} \\ |u(k+l) - u(k+l-1)| &\leq \Delta u_{max} \\ y_{min} &\leq y_m(k+l) \leq y_{max}. \end{aligned}$$

Model output $y_m(k+l)$ is calculated based on (2):

$$y_m(k+l) = \sum_{i=1}^{n_H} \omega_i \cdot \frac{1}{1 + e^{-(\sum_{j=1}^{n_u} u(k+l-j)\omega_{u_{ij}} + \sum_{j=1}^{n_y} u(k+l-j)\omega_{y_{ij}} + b_i)}} + b. \quad (11)$$

Taking into account the term $\alpha \sum_{m=1}^{N_c} \Delta u^2(k+l-1)$ in the cost function (10) prevents the control signals to be too big such that the executive organs would not be able to realize.

The control signals in the k -th step: $[u(k), u(k+1), \dots, u(k+N_c-1)]^T$, is possible to solve both numerically and analytically. Analytically it is solved in such a manner that from the system of algebraic equations:

$$\frac{\partial J}{\partial u(k)} = 0, \frac{\partial J}{\partial u(k+1)} = 0, \dots, \frac{\partial J}{\partial u(k+N_c-1)} = 0$$

one obtains $[u(k), u(k+1), \dots, u(k+N_c-1)]^T$.

For analytical solution, it is necessary for the model to be linear. The neural network model is nonlinear, its linearization should be performed first, or apply the numerical methods for solving the optimization problem. In this work for obtaining the optimum values of the control signals, the genetic algorithm is used (the function `ga` of the Matlab's Genetic Algorithm Toolbox). The genetic algorithms represent the global optimization technique. In the k -th step $N_c - 1$ control signals are obtained. The first calculated signal is being sent to the controller output.

The GA used in this study is simple genetic algorithm. The elements of the populations are encoded into bit-strings. The chromosome selection for reproduction is performed using the Roulette selection method. The multi-point crossover operator was used. The uniform mutation operator is applied in this study.

4 Simulation results

Example 1

For the simulation example 1, we consider the nonlinear plant, which is described by the following nonlinear difference equation:

$$y(k) = 0.5y(k-1) + u(k-1) \left(1 + 0.2y^2(k-1) \right) + u^3(k-1), \quad (12)$$

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

We assume that structure of the model is known, $n_u = 1, n_y = 1$. The inputs and output of the neural network model are $u(k-1), y_m(k-1)$ and $y_m(k)$ respectively. The data set are obtained by applying random input signal uniformly distributed in the interval $[-0.5, 0.5]$. The plant output is bounded within region $[-2.2, 2.2]$. In this example, 4500 training patterns are generated to train the DRN and 1500 to test the obtained DRN model.

Selection of an appropriate number of neurons in the hidden layer is very important. The optimal network size was selected from the one which resulted in maximum correlation coefficient for the training and test sets, Table 1.

Based on Table 1, it was concluded that the optimal number of hidden neurons is 12.

Table 1. Correlation coefficient for the training and test sets

DRN-structure	2-10-1	2-12-1	2-15-1	2-17-1
Training	0.9878	0.9987	0.9856	0.9879
Test	0.9845	0.9921	0.9795	0.9812

Since the object model is formed, it is necessary to define the cost function parameters, (10). These parameters are selected by the trial and error method. By increasing the variance of the control signal is decreasing, but simultaneously the difference between the set and real value of the object output is increasing. In the considered example, the satisfactory results are obtained for the following values of parameters: $N_H = 3$, $N_C = 3$, $\alpha = 0.05$.

The optimal predictive controller in the k -th step computes the control signals:

$$[u(k), u(k+1)]^T.$$

In Figure 4, the reference signal is shown. The difference between the reference signal and the object output is presented in Figure 5. From the Figure 5, it is obvious that the tracking error is small. The obtained solution is good for practical realization.

Example 2

The plant is given by:

$$y(k) = 0.35 \left\{ \frac{y(k-1)y(k-2)[y(k-1) + 2.5]}{1 + y^2(k-1) + y^2(k-2)} + u(k-1) \right\}, \quad (13)$$

where y is the output of the plant and u is the plant input. The dynamical system used in the simulation example is given in [21]. It is assumed that structure of the model is known, $n_u = 1$, $n_y = 2$. The inputs and output of the neural network are $u(k-1)$, $y_m(k-1)$, $y_m(k-2)$ and $y_m(k)$, respectively.

The input-output patterns are generated randomly. The data set included 4000 data samples. In the training process of the DRN, 3000 samples were used. The DRN model was tested using 1000 selected data. In this example, it is found that the optimal number of hidden neurons is 15 (Table 2).

Table 2. Correlation coefficient for the training and test sets

DRN-structure	3-10-1	3-12-1	3-15-1	3-17-1
Training	0.9811	0.9899	0.9945	0.9921
Test	0.9799	0.9831	0.9897	0.9895

The control and predictive horizons are chosen as respectively, $N_H = 3$, and $N_C = 3$. The choice of these values determines the complexity of the optimization problem. The values of N_H and N_C may not be too large, because the computation time would become also too large. These values should provide a good compromise between performance and computational load. It is found that the weight factor of the control signal $\alpha = 0.045$ by the trial and error method.

The optimal predictive controller in the k -th step computes the control signals: $[u(k), u(k+1)]^T$. The reference signal is shown in Figure 6. The difference between the reference signal and the object output is presented in Figure 7.

From the Figure 7, it is obvious that the control is quite effective. The obtained solution is good for practical realization.

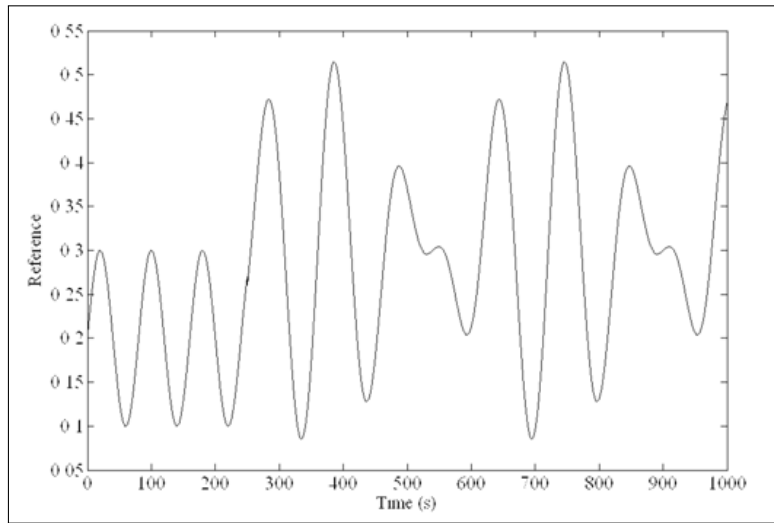


Figure 4: The reference signal

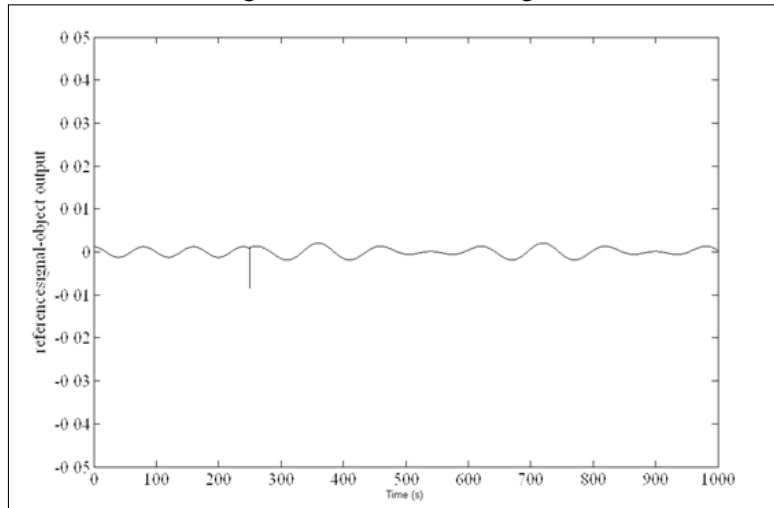


Figure 5: Difference between the reference signal and the object output

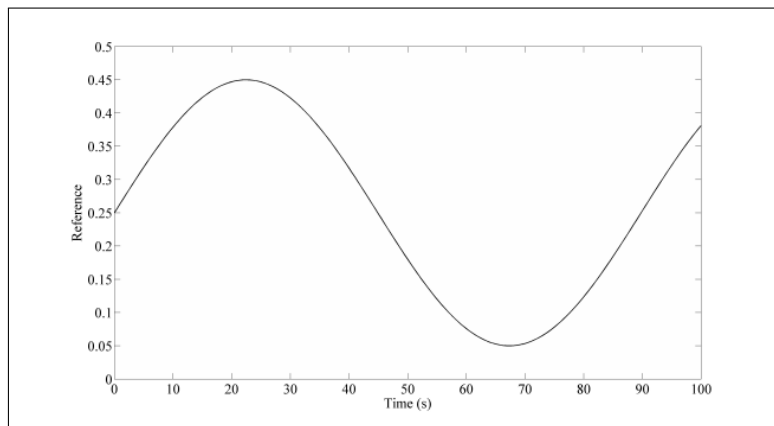


Figure 6: The reference signal

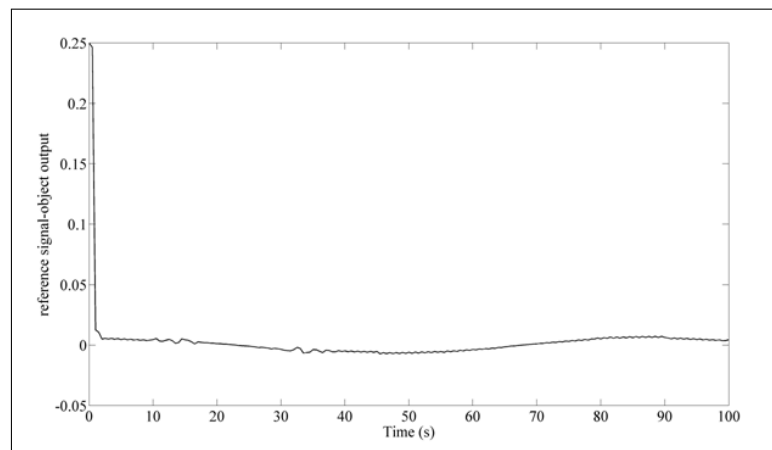


Figure 7: Difference between the reference signal and the object output

5 Conclusions

In this paper the synthesis of the predictive controller for control of the nonlinear object is considered. The object is modeled by the digital recurrent network. 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 large number of parameters (weights and biases) that need to be adapted.

The simulation results, given in Section 4, show that the predictive controllers can successfully be applied for control of the prominently nonlinear object. The optimum values of the control signals are obtained by the genetic algorithm which represents the global optimization technique. The given trajectory tracking error is small. The proposed structure can be applied in control of the linear objects that are modeled by the neural network. Recurrent networks are more powerful than nonrecurrent networks and have important uses in control and signal processing applications.

Simulation results on the examples selected from literature provide good results and that encourages our future plan for real time control system implementation.

Acknowledgement

Parts of this research were supported by the Ministry of Sciences, Technologies and Development of Republic of Serbia.

Bibliography

- [1] S. Jagannathan, F.L. Lewis, Identification of Nonlinear Dynamical Systems Using Multilayered Neural Networks, *Automatica*, 32(12):1707-1712, 1996.
- [2] G. Kenne, T. Ahmed-Ali, F. Lamnabhi Lagarrigue, H. Nkwawo, Nonlinear systems parameters estimation using radial basis function network, *Control Engineering Practice*, 14(7):819-832, 2006.
- [3] J.I. Canelon, L. Shieh, N.B. Karayiannis, A new approach for neural control of nonlinear discrete dynamic systems, *Information Sciences*, 174(3-4):177-196, 2005.
- [4] W. Yu, Nonlinear system identification using discrete-time recurrent neural networks with stable learning algorithms, *Information Sciences*, 158(1):131-147, 2004.

- [5] S.-K. Oh, W. Pedrycz, H.-S. Park, Hybrid identification in fuzzy-neural networks, *Fuzzy Sets and Systems*, 138(2):399-426, 2003.
- [6] A. Yazdizadeh, K. Khorasani, Adaptive time delay neural network structures for nonlinear system identification, *Neurocomputing*, 47(1-4):207-240, 2002.
- [7] M. Onder Efe, O. Kaynak, A comparative study of neural network structures in identification of nonlinear systems, *Mechatronics*, 9(8):287-300, 1999.
- [8] V. Ranković, I. Nikolić, Identification of Nonlinear Models with Feedforward Neural Network and Digital Recurrent Network, *FME Transactions*, 36(2):87-92, 2008.
- [9] S. Dubljevic, P. Mhaskar, N.H. El-Farra, P.D. Christofides, Predictive control of transport-reaction processes, *Computers and Chemical Engineering*, 29(11-12):2335-2345, 2005.
- [10] P. Mhaskar, N.H. El-Farra, P.D. Christofides, Robust hybrid predictive control of nonlinear systems, *Automatica*, 41(2):209-217, 2005.
- [11] H. Peng, T. Ozaki, Y. Toyoda, K. Oda, Exponential ARX model-based long-range predictive control strategy for power plants, *Control Engineering Practice*, 9(12):1353-1360, 2001.
- [12] V. Ranković, I. Nikolić, Model Predictive Control Based on the Takagi-Sugeno Fuzzy Model, *Journal of Information, Control and Management Systems*, 5(1):101-110, 2007.
- [13] M. Lazar, O. Pastravanu, A neural predictive controller for non-linear systems, *Mathematics and Computers in Simulation*, 60(3-5):315-324, 2002.
- [14] C.-H. Lu, C.-C. Tsai, Generalized predictive control using recurrent fuzzy neural networks for industrial processes, *Journal of Process Control*, 17(1):83-92, 2007.
- [15] K. Laabidi, F. Bouani, M. Ksouri, Multi-criteria optimization in nonlinear predictive control, *Mathematics and Computers in Simulation*, 76(5-6):363-374, 2008.
- [16] M. Hagan, O.D. Jesus, R. Schultz, Training Recurrent Networks for Filtering and Control, Chapter 11 of *Recurrent Neural Networks: Design and Applications*, L.R. Medsker and L.C. Jain, Eds., CRC Press, 325-354, 1999.
- [17] R.K. Al Seyab, Y. Cao, Nonlinear system identification for predictive control using continuous time recurrent neural networks and automatic differentiation, *Journal of Process Control*, 18(6):568-581, 2008.
- [18] D.R. Hush, B.G. Horne, Progress in supervised neural networks, *IEEE Signal Processing Magazine*, 10(1):8-39, 1993.
- [19] K.L. Funahashi, Y. K.L. Funahashi, Y. Nakamura, Approximation of dynamical systems by continuous time recurrent neural networks, *Neural Networks*, 6(6):183-192, 1993.
- [20] L. Jin, P. Nikiforuk, M. Gupta, Approximation of discrete-time state-space trajectories using dynamic recurrent neural networks, *IEEE Transactions on Automatic Control*, 40(7):1266-1270, 1995.
- [21] P.S. Sastry, G. Santharam, K.P. Unnikrishnan, Memory Neuron Networks for Identification and Control of Dynamical Systems, *IEEE Transactions on Neural Networks*, 5(2):306-319, 1994.