NEURAL CONTROLLERS FOR NONLINEAR SYSTEMS IN MATLAB

S.Kajan
Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Slovak Republic

Abstract

This paper deals with design of neural controllers for nonlinear systems control. For the purpose of neural control structures a direct and inverse neural model of a nonlinear dynamic system using three-layer perceptron network was created. These neural models were used in following control structures: direct inverse control, internal model control and predictive control. The performance tests for particular controllers were realized in the simulation environment Matlab/Simulink using selected types of nonlinear dynamic processes.

1 Introduction

For purposes of nonlinear system control, it is important to have accurate models. Thanks to a very good approximating ability of multi-layer perceptron networks (MLP) we are able to create accurate neural models of nonlinear processes. For purpose of control of nonlinear dynamic systems, several control structures using neural models and inverse neural models have been developed, and this article deals with them.

2 Neural models

Neural model of process is represented by three-layer artificial neural network of MLP type. The objective of MLP network is to approximate the relation of system output in *k*-th step on basis of past values of system output and input and thus get feed-forward neural model. Than we can describe nonlinear dynamic system by following model:

$$\hat{y}(k+1) = f[y(k), y(k-1), \ldots, y(k-n+1), u(k), u(k-1), \ldots, u(k-m+1)] \tag{1}$$

where u- process input, y- process output, n- order of process output , m- order of process input, f- nonlinear function, k- discrete time ($t = k * T_{vz}$, T_{vz} is sampling period).

For purpose of system control, we used inverse neural model, which we get by exact inversion of model from equation (1) by expressing controller output u(k):

$$u(k) = f^{-1}[\hat{y}(k+1), y(k), y(k-1), \ldots, y(k-n+1), u(k-1), \ldots, u(k-m+1)] \tag{2}$$

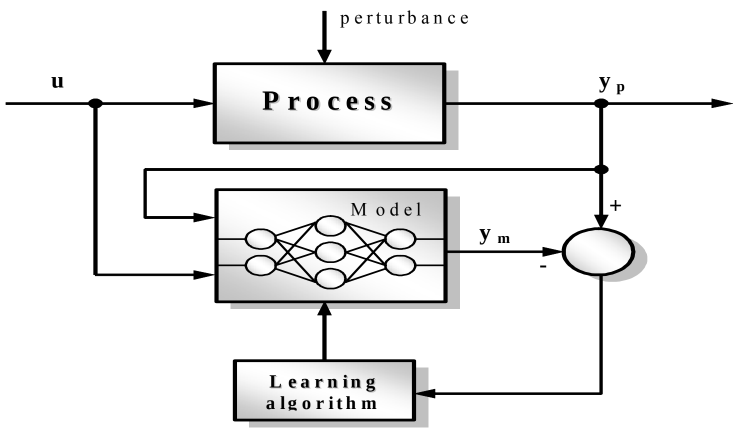


Fig. 1 The process modelling block scheme using artificial neural network

A block scheme of the artificial neural network process model is in Fig. 1. [1]. The neural model is located parallel to process, and prediction error is used as network training signal for the learning algorithm. The Levemberg-Marquart method has been used for training the MLP network [2].

3 Neural controllers for nonlinear systems

Neural controllers for control of nonlinear processes are using inverse neural models. We've chosen following neural controllers to compare the control performance of nonlinear processes:

- Robust direct inverse neural control [4],
- Internal model control with neural models [1],
- Predictive neural control [3, 5, 6]

3.1 Robust direct inverse neural control

In direct inverse control, inverse model described by equation (2) is used. We used predicted value $\hat{y}(k+1)$ known from the set of input-output data while training the inverse model. In closed-loop circuit the predicted value $\hat{y}(k+1)$ is replaced by reference value $w(k+1)$, and thereby we get closed-loop neural controller out of the inverse model. In Fig.2 is shown a block scheme of direct inverse control.

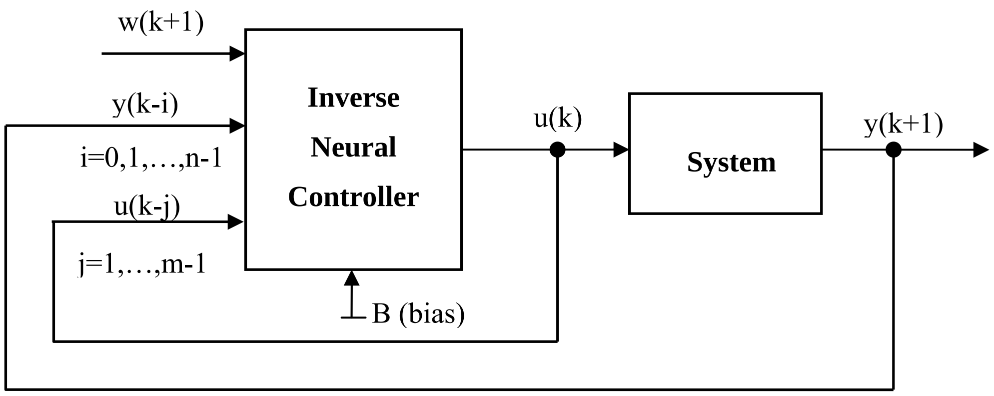


Fig. 2 The scheme of direct inverse neural control

Direct inverse control as shown in Fig.2 can not remove permanent reference error when the system parameters changed, or disruption occurs. Therefore an adapting block which adapts neuron threshold value in output layer of neural network is added to control. [4].

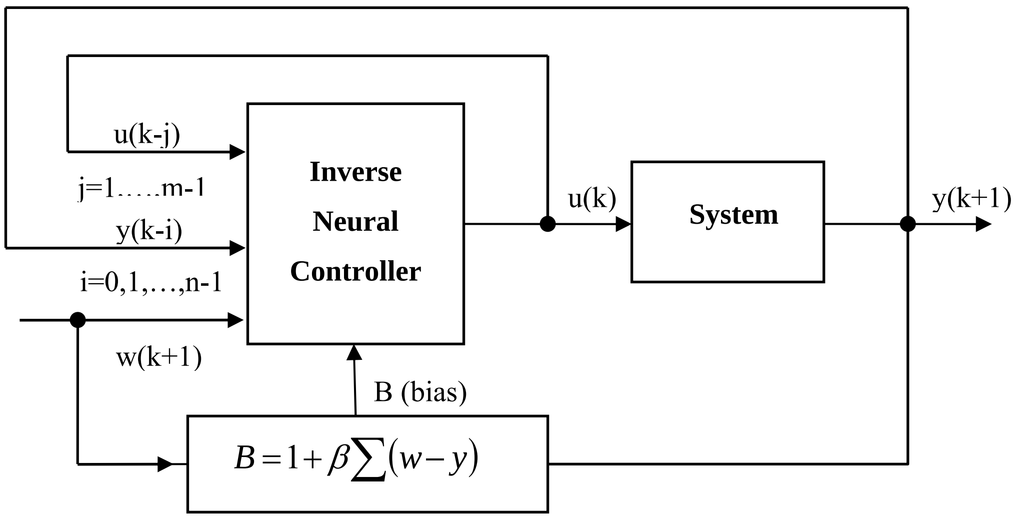


Fig. 3 The scheme of robust direct inverse neural control

In Fig.3 is displayed a block scheme of robust inverse control, where the adapter is as a simple integrator in form:

$$B = 1 + \beta \sum (w - y) \tag{3}$$

where β is adaptive parameter from range 0 to 1.

3.2 Internal model control with neural models

The IMC control structure uses inverse neural model of system as a controller. It uses negative feedback of difference between the system output and output of neural model to suppress the reference error. (see Fig.4). A filter to attenuate step changes of differences may be connected in the negative feedback.

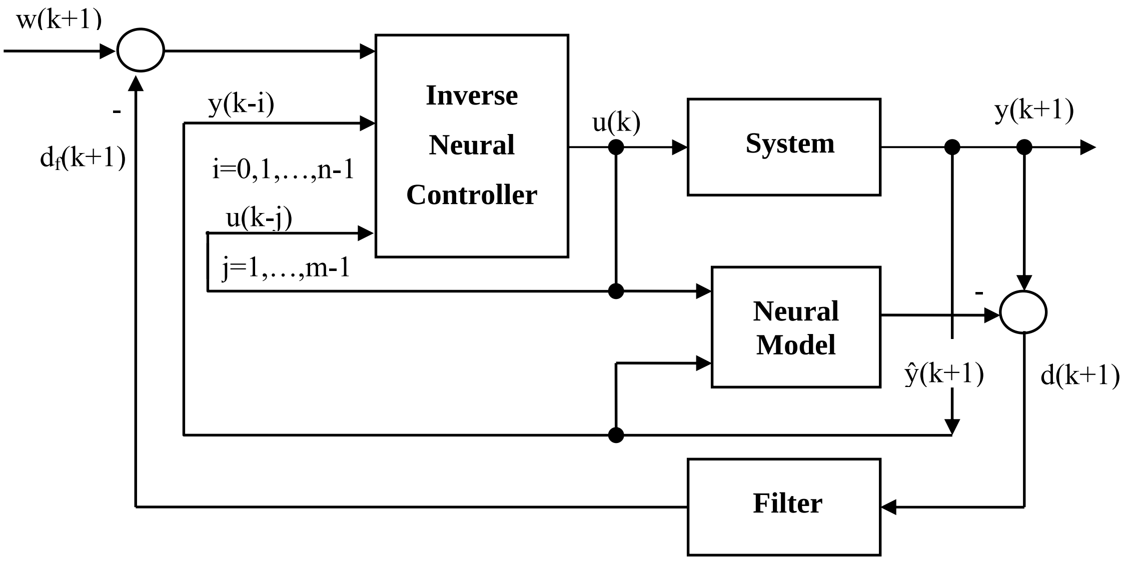


Fig. 4 The scheme of internal model control with neural models

3.3 Predictive neural control

Connection of this control structure is shown in Fig.5. It uses direct neural model to predict future outputs of process assuming the control variable will be u'. This control variable is optimized in each step of control process, so that predicted value of output $\hat{y}(k+i)$ step reaches reference value *w*.

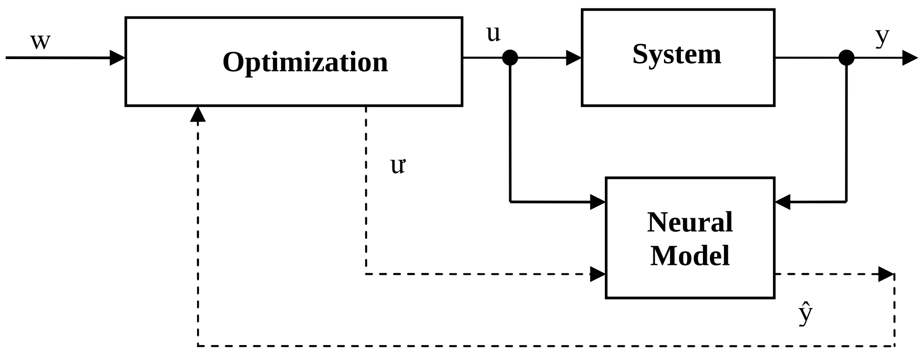


Fig. 5 The scheme of predictive neural control

4 Simulation results

Testing of control quality of selected nonlinear systems with several types of neural controllers was realized in simulation environment of Matlab Simulink. For the purpose of testing, we used simulation models of nonlinear dynamic systems described by following differential equations, where y is output and u is input of the system:

System A). $y''+0.7y'+0.2y+0.3y^3-u=0$ (3)

System B). $y''+y'(y^2+1).0.03+0.1y$^2-0.1u=0 (4)

System C). $y''+5y'+3y+2y^3-3u(1+y)=0$ (5)

Listed nonlinear systems have nonlinear transfer characteristic and dynamics of the system changes according to operating point, where the range of system input is 0 to 10. In Matlab environment, we used simulation models of systems to generate training and testing data to create neural and inverse neural model of system, described by equations (1) and (2), where we put *m* and *n* parameters equal 2. Neural model was created using Neural Toolbox, where we used MLP network with one hidden layer with 9 neurons and *tansig* activation function for modelling. We used Levemberg-Marquart method for training of the MLP network [2]. We created a simulation scheme for each type of neural controller (see Fig. 6 to 9). Simulation scheme for predictive control has been created by modifying an existing scheme in Neural toolbox of Matlab [6]. For each system, we performed a simulation of time responses of control signal and system output for step changes of reference value. Time responses of system output and reference value of some systems for individual types of control are depicted in Figures 10a), 11a) and 12a). Time responses of system output and reference value of some systems with perturbation occurring in time t=30s and t=50s and value of 0.2 are depicted in Figures 10b), 11b) and 12b). Numerical comparisons of control quality criteria are shown in tables Tab 1., Tab 2. and Tab 3. We evaluated quality criteria like overshoot, control time and IAE – integral of absolute values of control error.

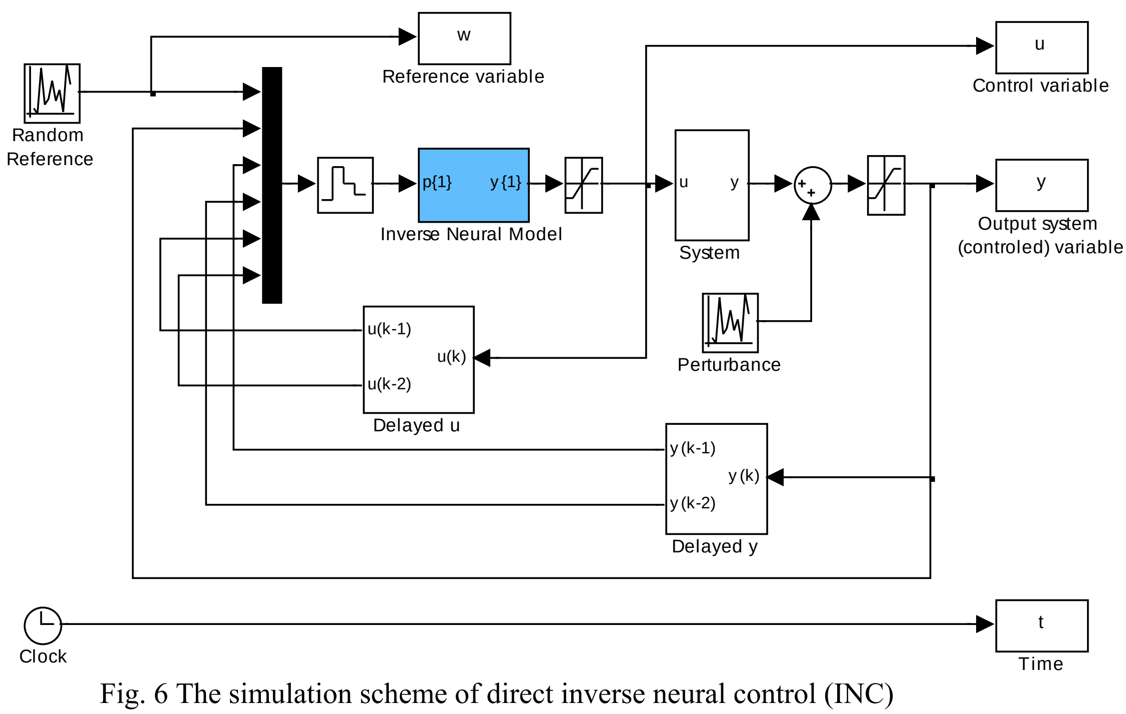


Fig. 6 The simulation scheme of direct inverse neural control (INC)

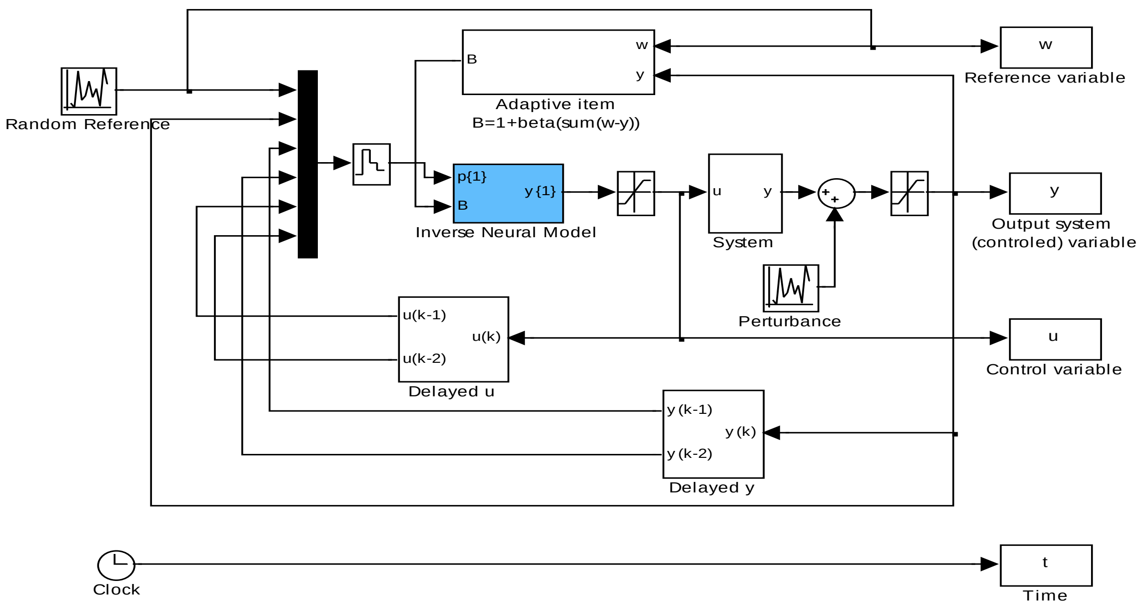


Fig. 7 The simulation scheme of robust direct inverse neural control (RINC)

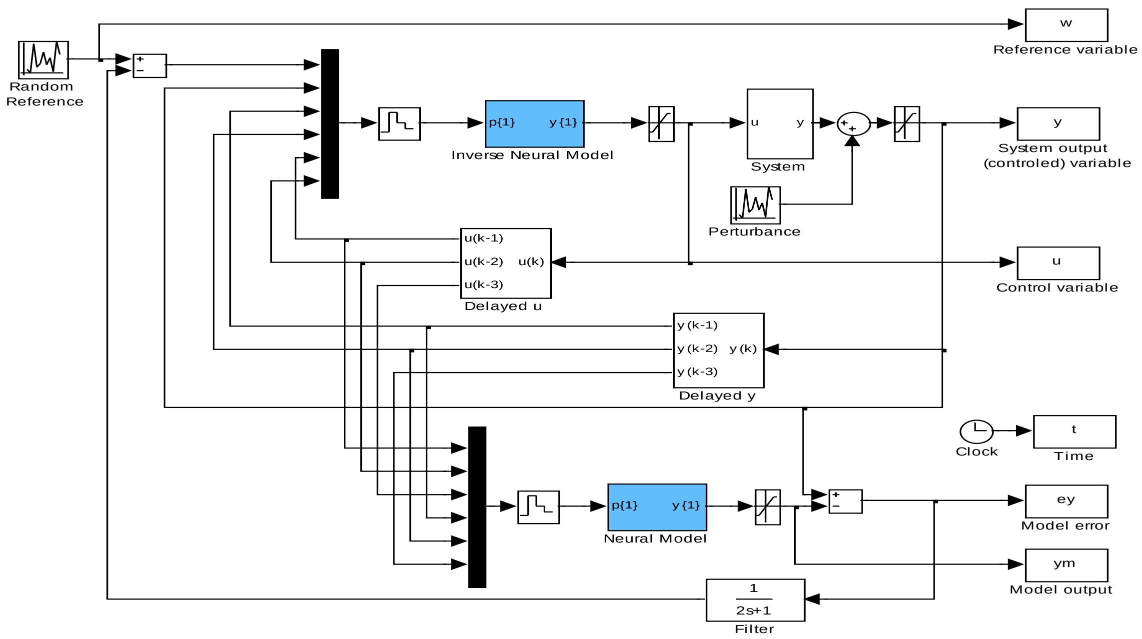


Fig. 8 The simulation scheme of internal model control with neural models (IMNC)

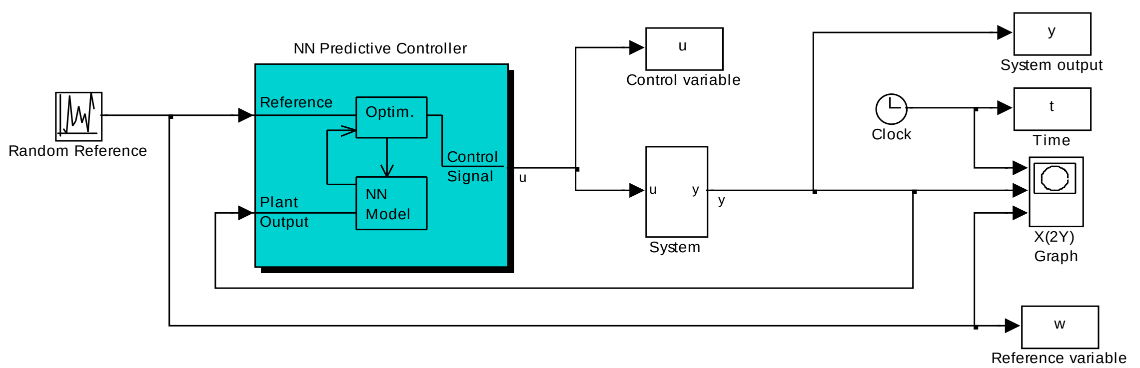
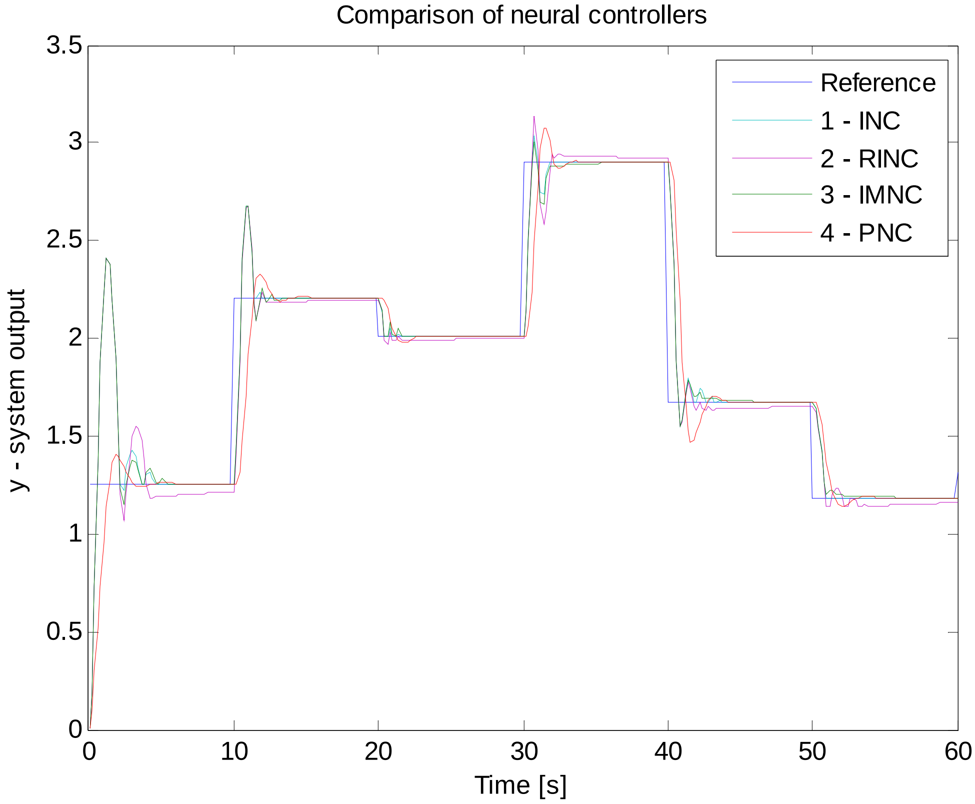
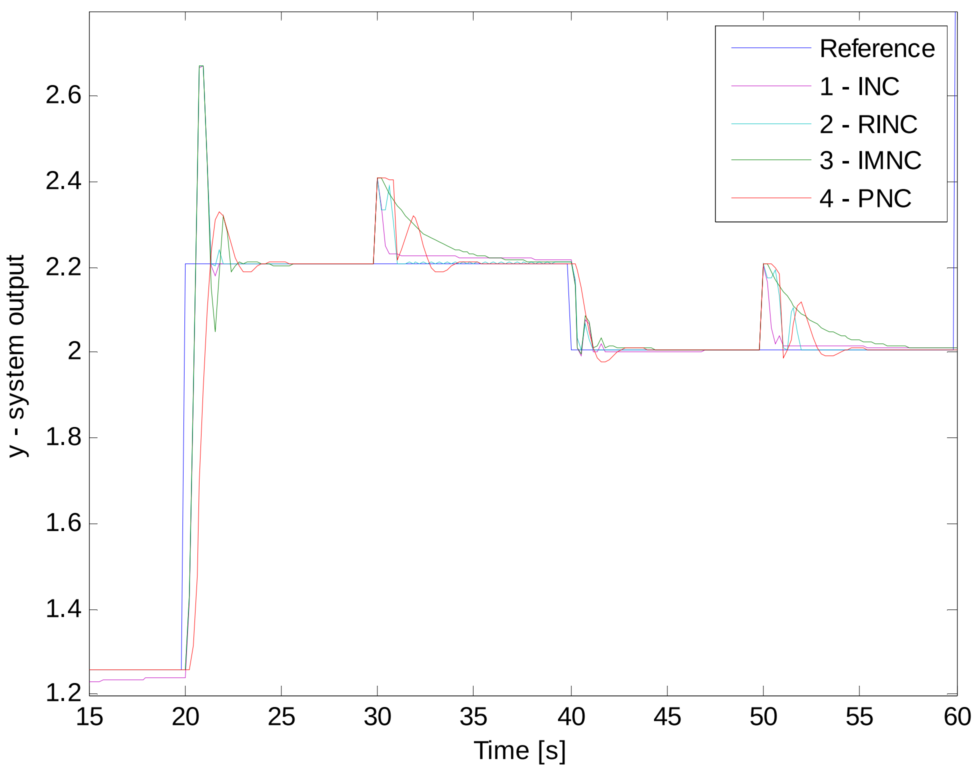


Fig. 9 The simulation scheme of predictive neural control (PNC)

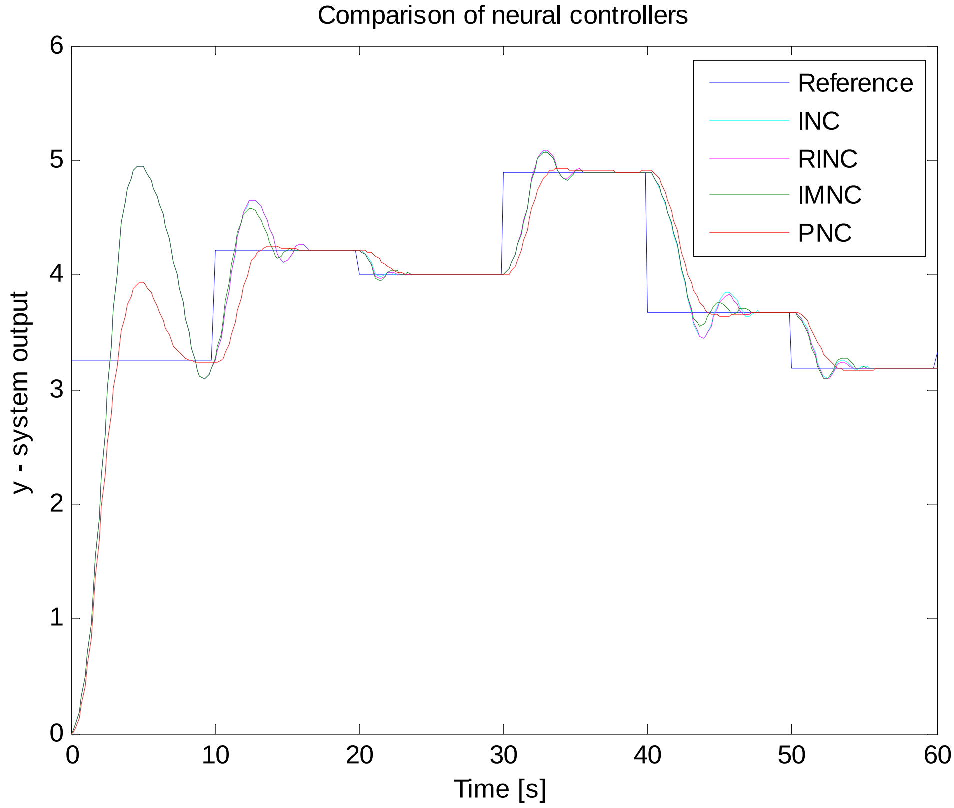


a)

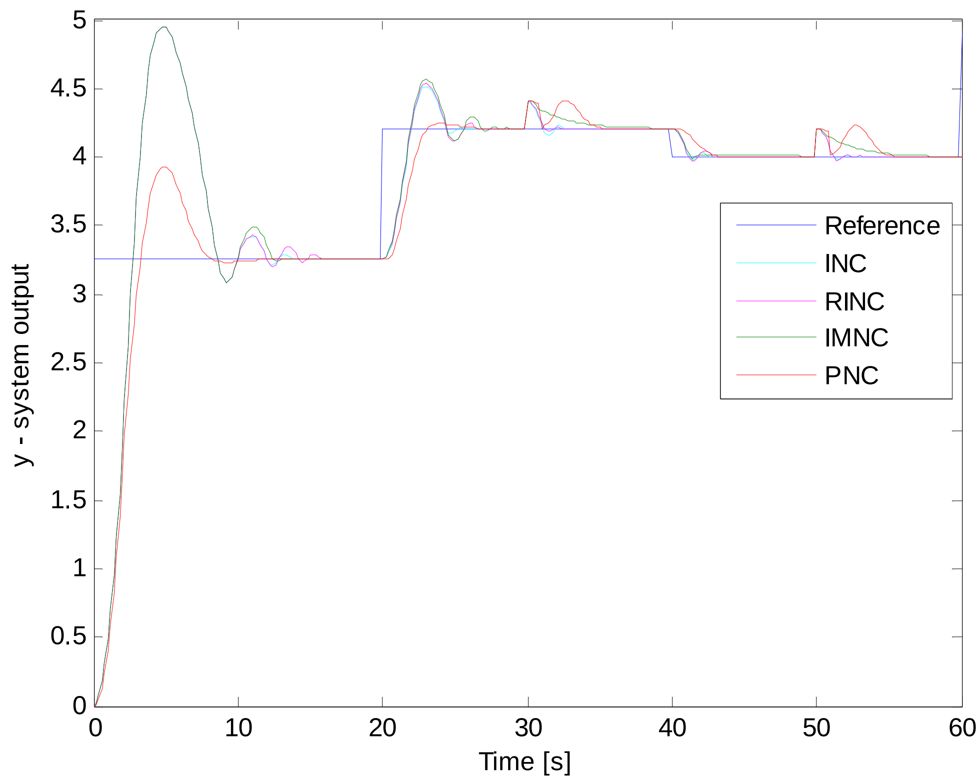


b)

Fig. 10 Time responses comparison of some neural controllers for A system a) with perturbation b)

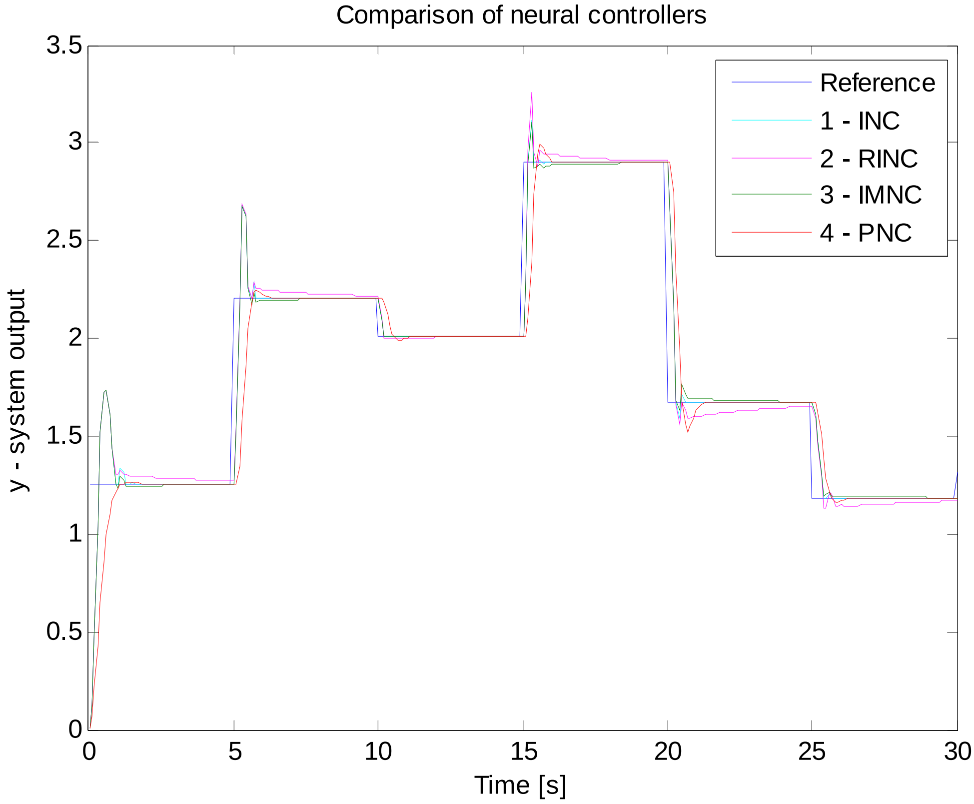


a)

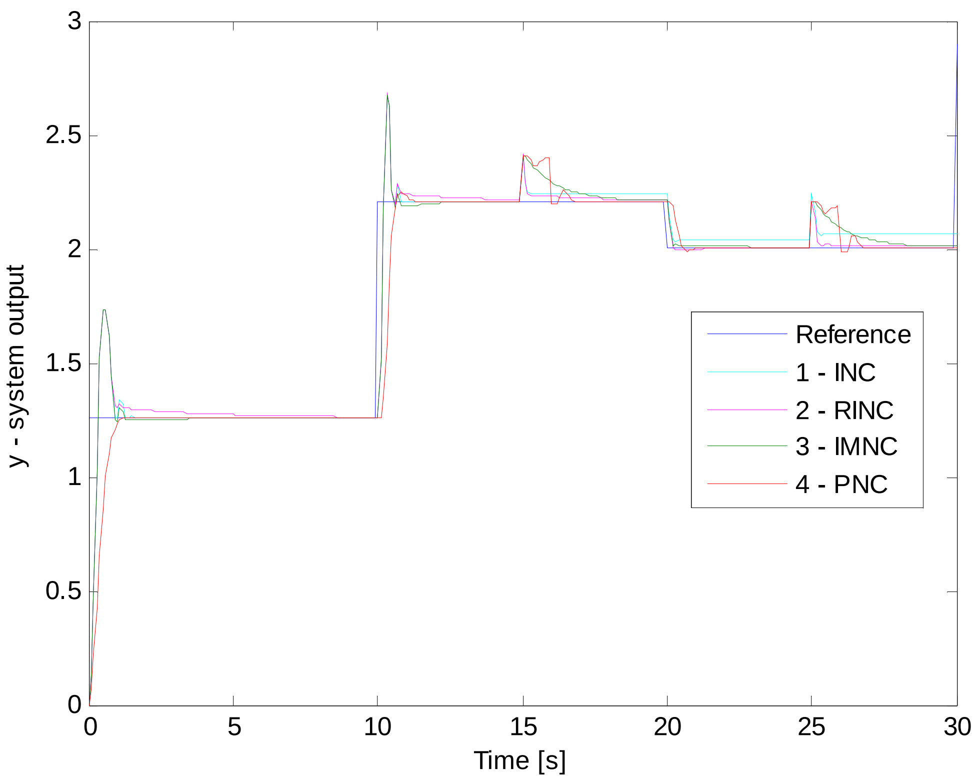


b)

Fig. 11 Time responses comparison of some neural controllers for B system a) with perturbation b)



a)



b)

Fig. 12 Time responses comparison of some neural controllers for C system a) with perturbation b)

Tab. 1 Control performance values for A system

	Without perturbation (Fig.10a)			With perturbation (Fig.10b)		
	IAE	overshoot [%]	control time [s]	IAE	overshoot [%]	perturbation. time [s]
INC	2,38	0 – 39	1 – 3,2	1,16	10 – 10,5	1,5 - 3
RINC	3,55	5,2 – 36,9	5 – 9,6	1,21	10 – 10,5	4,2 – 7,5
IMNC	2,65	0 - 39	2,4 – 2,8	1,85	10,2 - 11	5,2 – 8,4
PNC	3,89	5,6 – 15,3	3 – 3,6	1,84	10,2 - 11	3,6 – 4,2

Tab. 2 Control performance values for B system

	Without perturbation (Fig.11a)			With perturbation (Fig.11b)		
	IAE	overshoot [%]	control time [s]	IAE	overshoot [%]	perturbation. time [s]
INC	7,3	4,2 – 33,4	2,4 – 7,2	2,35	10,5 - 11	2,1 – 2,4
RINC	7,31	5,3 – 33,6	2,6 – 7	2,47	10 – 10,5	2,3 – 2,8
IMNC	6,78	5,6 – 26,2	3,6 – 7	3,16	10 – 10,5	5,2 – 6,8
PNC	7,77	1,5 – 3,1	4,6 – 6	2,95	10,2 - 11	5,2 – 5,8

Tab. 3 Control performance values for C system

	Without perturbation (Fig.12a)			With perturbation (Fig.12b)		
	IAE	overshoot [%]	control time [s]	IAE	overshoot [%]	perturbation. time [s]
INC	0.89	0 – 38,9	0.2 – 0,8	0,9	10,5 - 11	---
RINC	1,42	1,5 – 41,2	0.4 – 4,3	0,53	10 – 10,5	1,2 – 2,8
IMNC	1,05	0,2 – 38,45	0,2 – 1,2	0,81	10,2 - 11	2,8 – 4,2
PNC	1,56	2,3 – 7,2	1 – 1,2	0.89	10,5 - 11	1,8 – 2,3

5 Conclusion

The main objective of this article was to compare particular structures of neural controllers on selected types of nonlinear processes. We used Matlab Simulink to evaluate quality control criteria and behaviour of particular values to generalize for particular types of neural controllers.

Acknowledgment

The work has been supported by the grant agency VEGA No. 1/3100/06.

References

[1] A. Jadlovská. *Modelovanie a riadenie dynamických procesov s využitím neurónových sietí*. Edícia vedeckých spisov FEI TU Košice, ISBN 80-8894122-9, 2003

[2] M. T. Hagan, M. B. Menhaj. *Training Feedforward Networks with the Marquardt Algorithm.* Submitted to the IEEE Proceedings on Neural Ne, 1994

[3] R. Keyser, O. Pastravanu, D. Onu. *MATLAB Neural Network Toolbox based software environment for nonlinear identification and control*, IFAC Workshop New Trends in Design of Control Systems, Smolenice, Slovakia, 1994

[4] L. Pastoreková, A. Mészáros, P. Burian. *Inteligentné riadenie systémov na báze inverzných neurónových modelov*. ATP journal Plus7, 2005, pp. 53-58

[5] S. Kajan, Š. Kozák. *Adaptive predictive control of nonlinear dynamical systems using neural networks*, 2nd International Prediction Conference, Nostradamus '99, October Zlin 1999, pp.32-37

[6] The Mathworks. *Neural Network Toolbox,User's Guide*, 2002

Ing. Slavomír Kajan, PhD.:
Institute of Control and Industrial Informatics, FEI STU in Bratislava,
E-mail: slavomir.kajan@stuba.sk