Slovak University of Technology in Bratislava
Institute of Information Engineering, Automation, and Mathematics

PROCEEDINGS

17^{th} International Conference on Process Control 2009

Hotel Baník, Štrbské Pleso, Slovakia, June 9 – 12, 2009

ISBN 978-80-227-3081-5

http://www.kirp.chtf.stuba.sk/pc09

Editors: M. Fikar and M. Kvasnica

Sekaj, I.: Design of the Neural Model Structure Based on Genetic Algorithms, Editors: Fikar, M., Kvasnica, M., In *Proceedings of the 17th International Conference on Process Control '09*, Štrbské Pleso, Slovakia, 273–276, 2009.

Full paper online: http://www.kirp.chtf.stuba.sk/pc09/data/abstracts/043.html

DESIGN OF THE NEURAL MODEL STRUCTURE BASED ON GENETIC ALGORITHMS

S. Kajan*, I. Sekaj, Z. Dideková*****

Institute of Control and Industrial Informatics,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Ilkovičova 3, 812 19 Bratislava
*e-mail : * slavomir.kajan@stuba.skT, ** ivan.sekaj@stuba.sk,*
**** zuzana.didekova@stuba.sk*

Abstract: Genetic algorithm based neural model structure design of a non-linear dynamic system is described. The genetic algorithm represents an optimisation procedure, where the cost function, which is minimized consists of the non-linear dynamic process neural model simulation and a selected performance index evaluation. Using this approach the neural model of the process has been optimised from point of view its internal architecture. A multilayer perceptron (MLP) artificial neural network has been used, where the training was realized by the Levenberg-Marquardt method.

Index Terms: genetic algorithm, neural network model, architecture optimisation, multilayer perceptron network.

1 INTRODUCTION

To control of some classes of non-linear dynamic systems neuro-predictive controllers with advantage are used. Such control structures consist of neural model, which uses the system output prediction. In the presented project the system model is realised using artificial neural network. For the non-linear process modelling the MLP network is used, where the network is trained using measured input and output process data. Our aim is to find the optimal neural network structure of the process model with respect to some predefined maximal model error. For that reason an evolution-based approach has been used. Evolutionary search/optimisation approaches are able to construct new control laws and non-intuitive solutions as well. One of the most frequently used evolutionary techniques is the genetic algorithm (GA). Recently, genetic algorithms have been applied in the area of process modelling and control for solving a wide spectrum of various optimisation problems in several ways and with several aims. In the proposed project the GA's are used for search for the optimal neural model architecture.

2 NON-LINEAR PROCESS MODELLING WITH NEURAL NETWORK

Consider a non-linear continuous-time dynamic process with two inputs and one output, which is described by the non-linear difference equation

$$y(k)=f(y(k-1),...,y(k-m); u(k-1),...,u(k-n)) \quad (1)$$

where m and n are orders of $y(k)$ and $u(k)$, respectively, f is an unknown nonlinear function determined on-line by the Back-propagation (BP) algorithm.

The nonlinear function f can be approximated by a three-layer back-propagation neural network. In our case sigmoidal activation function in the hidden layer and linear activation function in output layer have been used (Čapkovič, M., Kajan, S. 2000). For the full interconnection of neurons a multilayer perceptron network with 6 inputs and 10 neurons in hidden layer has been used (Fig. 1). The process modelling block scheme using artificial neural network is in Fig. 2. For off-line training the learning algorithm performs the adaptation of neural network weights. The Levenberg-Marquardt algorithm has been used (Hagan, M.T., Menhaj M.B. 1994).

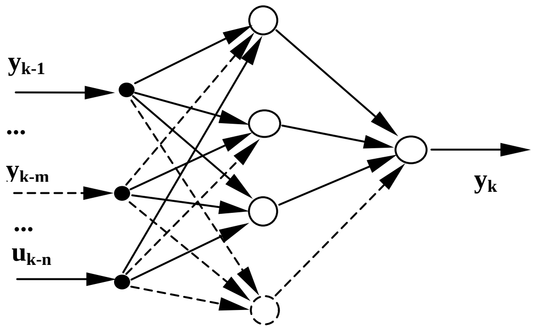

Fig. 1. Neural network structure for process modelling

Training of the neural network was realised using measured data (Fig.3) using the training function *trainlm* from the Neural Network Toolbox of Matlab. Cost function in form of the mean-squared error between the measured and model output (MSE) using training data has been minimized thanks modification of the network weights.

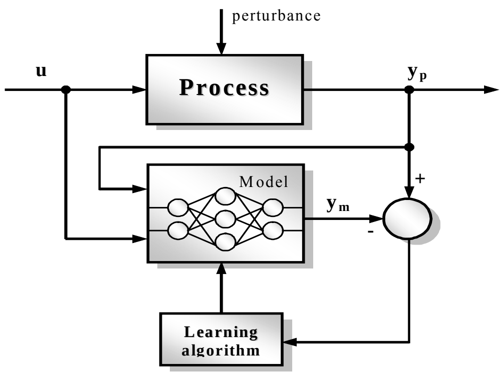

Fig. 2. Process modelling block scheme using artificial neural network

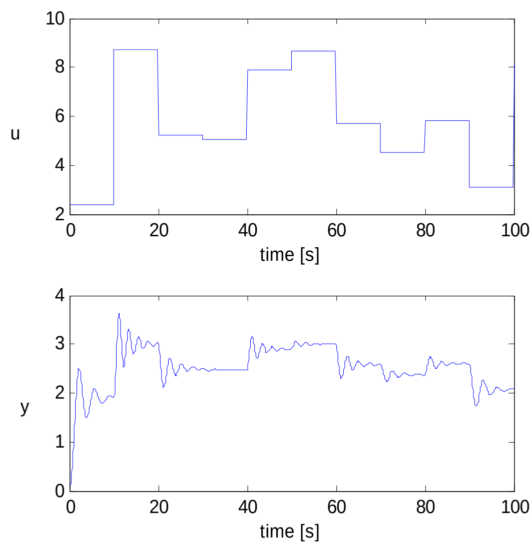

Fig. 3. Measured input/output process data for training of the neural network

3 GENETIC ALGORITHM

Genetic algorithm is a powerful stochastic-based search/optimisation approach, which mimics the evolution in the nature. It is described in e.g. (Goldberg, D.E. 1989, Man, K.F. Tang, K. S. Kwong, S. 2001, Michalewicz, Z. 1996, Sekaj, I. 2005) and others. A general scheme of a GA can be described by the following steps (Fig.4):

1. Initialisation of the population of chromosomes.

2. Evaluation of the cost function for all chromosomes.

3. Selection of parent chromosomes.

4. Crossover and mutation of the parents → children.

5. Completion of the new population from the new children and selected members of the old population. Jump to the step 2.

Before each cost function evaluation, the corresponding chromosome (genotype) is decoded into network interconnections of the neural model (phenotype) and after the simulation the performance index is evaluated.

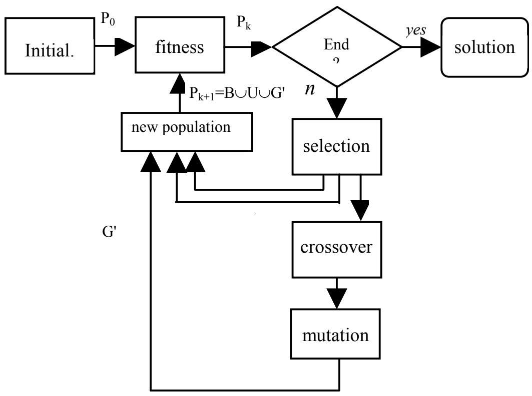

Fig. 4. Block scheme of the used genetic algorithm

4 CASE STUDY

For the demonstration of the proposed methods let us consider the non-linear system, which is described by the differential equation

$$y'' + 0{,}7y' + 0{,}2y + 0{,}3y^3 - u = 0\,. \qquad (2)$$

The simulation scheme for the described non-linear system (2) is depicted in Fig. 5. For modelling of the non-linear system the neural model structure with 6 inputs and 10 hidden neurons have been used (Fig. 6). For neural network training the measured data depicted in Fig. 3 have been used. The aim was to find the number of neurons in input and hidden layer and their interconnections.

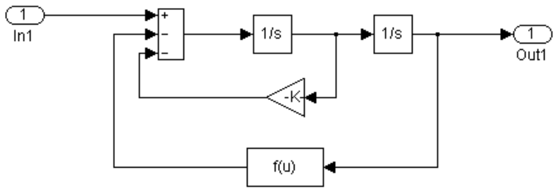

Fig. 5. Simulation block scheme of non-linear system

The interconnection map of the net is coded into the chromosome of the GA (Tab 1). The length of the chromosome is 70 genes. The gene coding is binary, where the value 1 represents connection between neurons, 0 represents no connection (Tab. 2).

	Input Neurons					Hidden Neurons					ON
	1	2	.	5	6	7	8	.	15	16	17
1	0	0	.	0	0	0	0	.	0	0	0
2	0	0	.	0	0	0	0	.	0	0	0
.	.	.	.	.	.	.	.	.	.	.	.
5	0	0	.	0	0	0	0	.	0	0	0
6	0	0	.	0	0	0	0	.	0	0	0
7	1	1	.	1	1	0	0	.	0	0	0
8	1	1	.	1	1	0	0	.	0	0	0
.	.	.	.	.	.	.	.	.	.	.	.
15	1	1	.	1	1	0	0	.	0	0	0
16	1	1	.	1	1	0	0	.	0	0	0
17	0	0	.	0	0	1	1	.	1	1	0

Tab.1: The interconnection map of neural network between neurons

1-10		11-20		.	41-50		51-60		61-70	
1	1	1	1	.	1	1	1	1	1	1

Tab.2: Chromosome from the interconnection map of the neural network

Genetic operations crossover and mutation are used for the chromosome modification. Note, that after crossover and mutation unrealizable neural network structures can be created. Hence, in case if an input neuron has no connection or a hidden neuron has no connection to input or output, such neurons are omitted from the net structure.

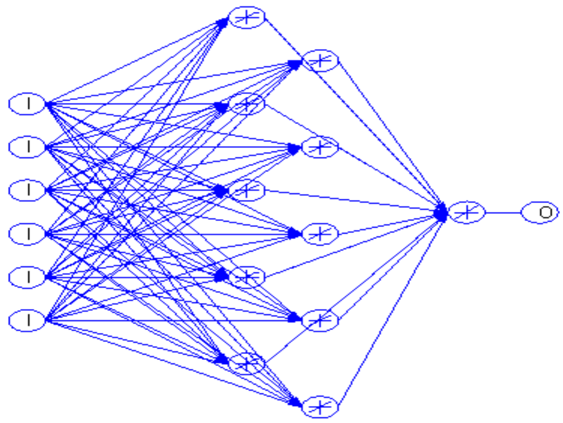

Fig. 6. Neural model structure with full connections and all neurons

For searching for optimal neural model structure the cost function (fitness) was considered in the form

$$F = MSE + (1-\alpha)*(w/w_n)*10^{-5} + \alpha*(n/n_n)*10^{-5} \quad (3)$$

MSE – mean square error of the neural model output (error between model and process output)
α – weight constant, which was set to 0,7
w – number of weighted interconnections between the input and hidden layer
w_n – maximum number of all interconnections
n – number of neurons in the network
n_n – maximum number of neurons in the network
10^{-5} – weight constant

The cost function evolution is shown in Fig. 9. The optimized neural network structure, which consists of 2 hidden neurons is depicted in Fig. 7. The obtained error using test data (Fig. 8), is MSE=0.4216e^{-6}.

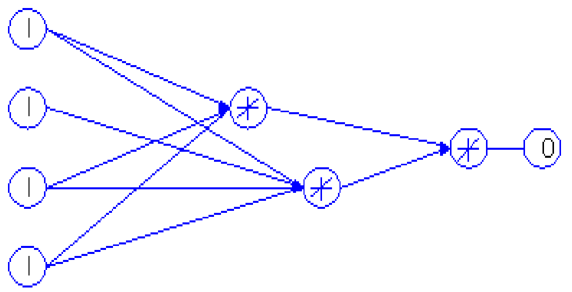

Fig. 7. Neural model with optimal structure designed by the GA

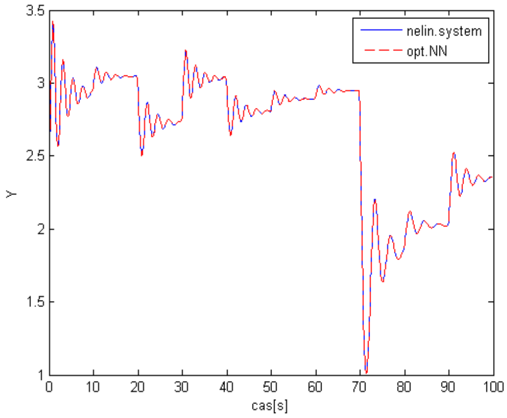

Fig. 8. Comparison of neural model and system outputs

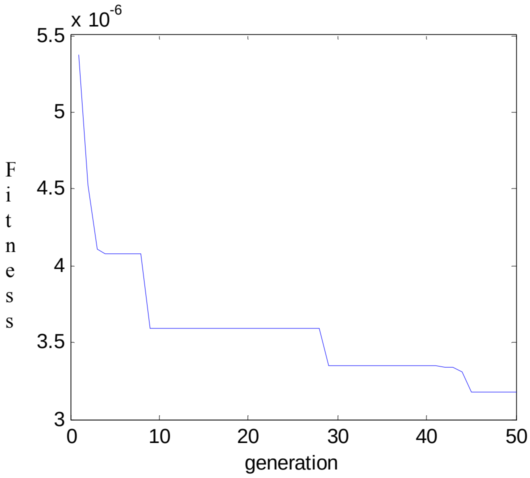

Fig. 9. Evolution of the fitness function

5 CONCLUSIONS

The aim of this project was to design the non-linear neural network model using genetic algorithm - based optimisation with focus on their architecture complexity reduction. The genetic algorithm represents an optimisation procedure, where the minimized cost function consists of the non-linear neural model simulation and the MSE performance index evaluation. The use of this approach leads to sufficient approximation accuracy of the MLP neural network with a considerable model complexity reduction degree.

ACKNOWLEDGMENTS

The work has been supported by the grants VEGA no. 1/0544/09 and no. 1/0690/09. This support is very gratefully acknowledged.

6 REFERENCES

Čapkovič, M., Kajan, S. (2000). Modelling of dynamical nonlinear systems using modular artificial neural network. In: *4th International Conference Control of Power Systems.* Bratislava

Demuth, H., Beale, M. (2003). Neural Network Toolbox, For use with Matlab*, User's guide*

Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimisation and Machine Learning. Addisson-Wesley.

Hagan, M.T., Menhaj M.B. (1994). Training Feedforward Networks with the Marquardt Algorithm. Submitted to the IEEE Proceedings on Neural Network

Man, K.F. Tang, K. S. Kwong, S. (2001). Genetic Algorithms, Concepts and Deign. Springer

Michalewicz, Z. (1996). Genetic Algorithms + Data Structures = Evolutionary Programs, Springer

Sartoris, P. (2008). Generovanie optimálnej štruktúry neurónových sietí pomocou genetických algoritmov, Diploma thesis, FEI STU Bratislava (in slovak)

Sekaj, I. (2003). Genetic Algorithm Based Controller Design", In: *2nd IFAC conference Control System Design'03*, Bratislava

Sekaj, I. (2005). Evolučné výpočty a ich využitie v praxi. Iris Bratislava (in slovak)