CONTROLLER PARAMETERS TUNING USING GENETIC ALGORITHM AND NEURAL MODEL

S. Kajan, M. Hypiusová

Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Slovak Republic
slavomir.kajan@stuba.sk, maria.hypiusova@stuba.sk

Abstract

The paper deals with a controller design for the nonlinear processes using genetic algorithm and neural model. The aim was to improve the control performance using genetic algorithm for optimal PID controller tuning. The plant model has been identified via an artificial neural network from measured data. The genetic algorithm represents an optimisation procedure, where the cost function to be minimized comprises the closed-loop simulation of the control process and a selected performance index evaluation. Using this approach the parameters of the PID controller were optimised in order to become the required behaviour of the control process. Testing of quality control process was realized in simulation environment of Matlab Simulink on 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. The aim was to improve the control performance using a genetic algorithm and neural model for optimal PID controller tuning written under MATLAB.

2 Modelling of nonlinear systems

Consider the neural model of a process is represented by a three-layer artificial neural network of the MLP type. The objective of the MLP network is to approximate the input/output relation of a system using the feed-forward neural model. We can describe a nonlinear dynamic system by the following model:

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

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

A block scheme of the artificial neural network process model is in Fig. 1. [4].

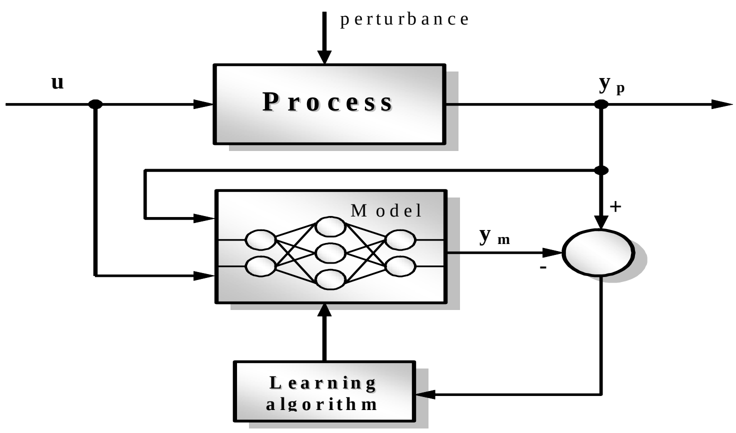

Figure 1: The process modelling block scheme using artificial neural network

The neural model is located parallel to process, and prediction error is used as network training signal for the learning algorithm. The Levenberg-Marquardt method has been used [3] for training the MLP network.

3 Controller design using genetic algorithm and neural model

As mentioned above, the aim of the control design is to provide required static and dynamic behaviour of the controlled process. Usually, this behaviour is represented in terms of the well-known concepts referred in the literature: maximum overshoot, settling time, decay rate, steady state error or various integral performance indices [1].

Without loss of generality let us consider a feedback control loop (closed-loop) (Fig.2), where y is the controlled value, u is the control value, w is the reference value and e is the control error $(e=w- y)$. Consider an appropriate simulation model using neural model of the controlled object is available [5].

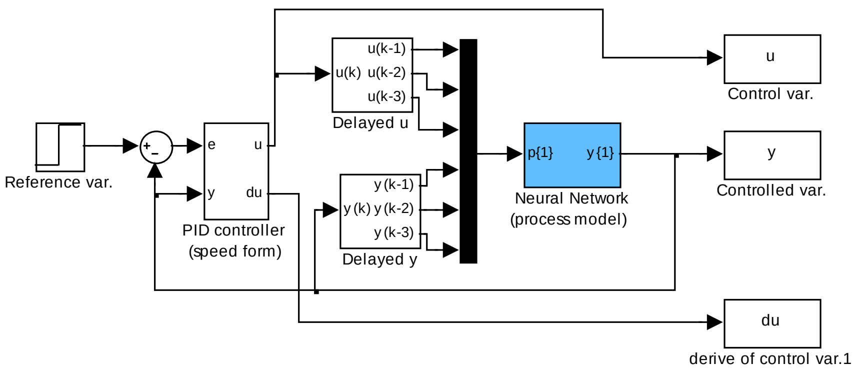

Figure 2: The Simulation scheme of feedback control loop.

Genetic algorithms are described in e.g. [2, 6, 7] and others. Each chromosome represents a potential solution, which is a linear string of numbers, whose items (genes) represent in our case the designed controller parameters. Because the controller parameters are real-number variables and in case of complex problems the number of the searched parameters can be large, real-coded chromosomes have been used.

Without loss of generality let us consider of a PID controller with feedforward speed structure, described in the discrete time domain by the equation (2), where P, I, D are controller parameters and T is sampling period, which was adjusted in value 0.1.

$$u(k) = u(k-1) + I.T.e(k) - P.\Delta y(k) - D.\frac{\Delta^2 y(k)}{T} \tag{2}$$

The searched controller parameters are $P \in R,\ I \in R,\ D \in R.$ The chromosome representation in this case can be in form ***ch***$=\{P,I,D\}$.

A general scheme of a GA can be described by following steps (Fig.3):

1. Initialization of the population of chromosomes (set of randomly generated chromosomes).

2. Evaluation of the cost function (fitness) for all chromosomes.

3. Selection of parent chromosomes.

4. Crossover and mutation of the parents $\rightarrow$ children.

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

The design procedure is based on the genetic algorithm (GA), which cost function (fitness) contains the closed-loop simulation with the neural model and the controller and the performance index evaluation (see Fig.4).

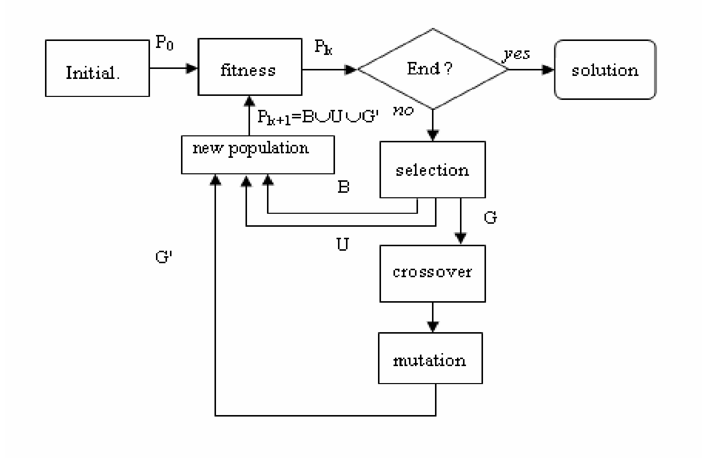

Figure 3: The Block scheme of genetic algorithm

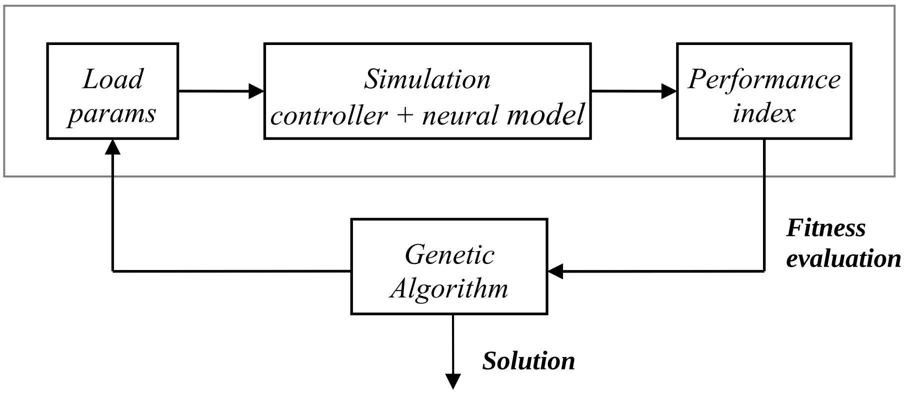

Figure 4: The Block scheme of the GA-based controller design

In domain of process control usually was used following cost function (fitness):

$$J = \sum_{i=1}^{N} |e_i| = \sum_{i=1}^{N} |w_i - y_i|, \tag{2}$$

where w is reference variable, y is controlled output, e is control error and N is number of patterns [8]. Fitness is represented by the cost function or in the case of control, by the modified cost function, which can be penalized for example by derivation of process output y, or by derivation of control action u, or by overshoot of controlled output y.

4 Examples of PID controller design using genetic algorithm and neural model

Testing of PID control quality of selected nonlinear systems 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 f_1, f_2, f_3 are nonlinear functions, 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). $f_1(y).y''+f_2(y).y'+y-f_3(y).u=0$ (4)

Listed nonlinear systems have nonlinear static characteristic and dynamics of the system changes according to operating point, where the range of system input is 0 to 10. Step responses of systems A and B for different of system input are depicted in Figure 5.

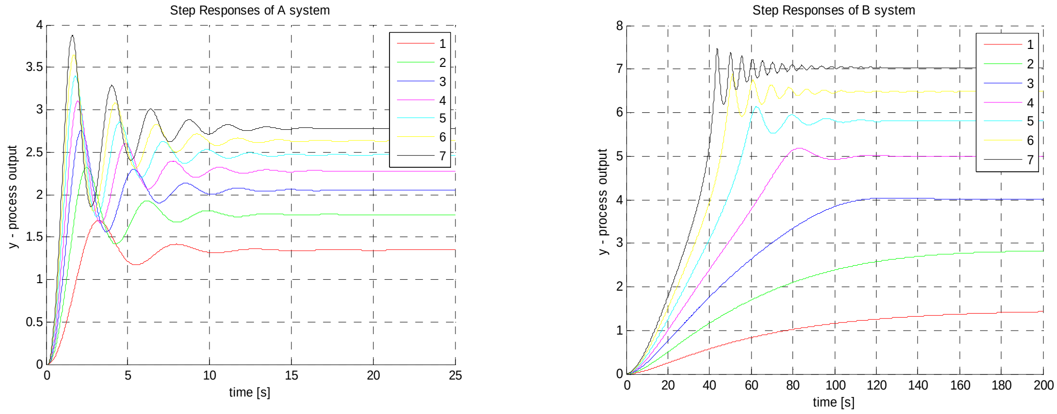

Figure 5: The Step responses of A and B systems

In Matlab environment, we used simulation models of systems to generate training and testing data to create neural model of system, described by equation (1), where we put *m* and *n* parameters equal 3. Neural model was created using Neural Toolbox, where we used MLP network with one hidden layer with 15 neurons and *tansig* activation function for modelling. We used Levenberg-Marquardt method [3] for training of the MLP network. The neural model was tested on the training and the testing data sets (Fig. 6). The deviation between the measured data and the neural network output obtained for training and testing data was 10^{-7}. The accurate evaluation of neural model errors is in Tab.1.

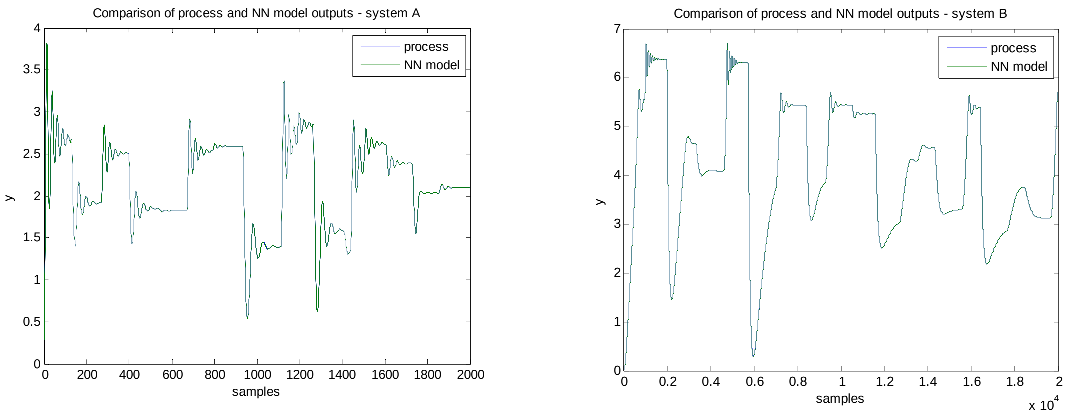

Figure 6: The Comparison of process and NN model outputs for A and B systems

System	Data	MSE $\frac{1}{N}\sum_{k=1}^{N}(y_p - y_m)^2$	SSE $\sum_{k=1}^{N}(y_p - y_m)^2$
A	Training	3.8659e-7	7.7318e-4
A	Testing	2.4968e-7	4.9936e-4
B	Training	3.3952e-9	6.7905e-5
B	Testing	7.9500e-7	1.59 e-2

Table 1: Neural model errors calculated for training and testing data

Consider that the GA finds the optimal (sub-optimal) solution in the user-defined search space of controller parameters, where these optimal PID controller parameter for A and B system are shown in Tab.2. For system A was used fitness function (2) with penalization by derivation of process output and for system B was fitness function extended any penalization by derivation of control action u and by overshoot of controlled output y. In Tab. 2 are presented also sum values of absolute values of control error (SAE) defined in equation (2). Time responses of the controlled variable, reference value and control variable for systems A and B are depicted in Fig. 7 and 8.

System	P	I	D	SAE
A	45.1526	68.8100	12.3780	11.0106
B	47.8107	3.8756	11.2250	440.5482

Table 2: The optimal PID controller parameter for A and B system

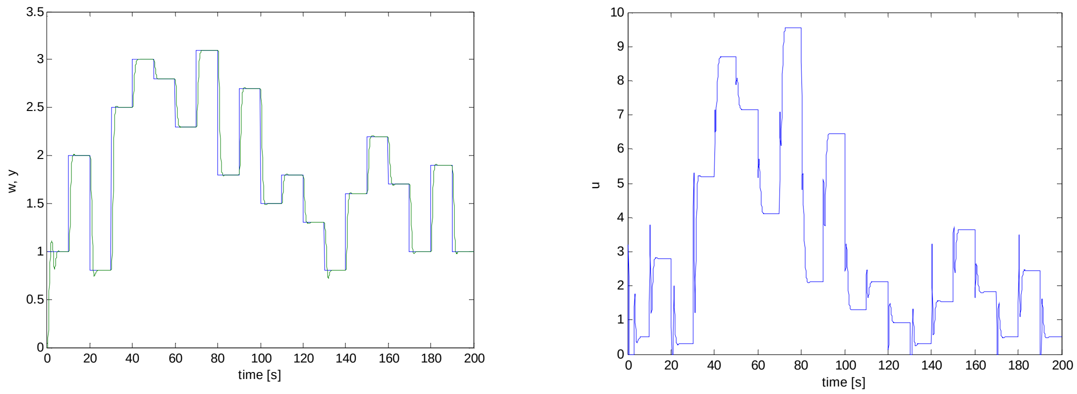

Figure 7: The Time responses of reference w, controlled y and control u variables for A system

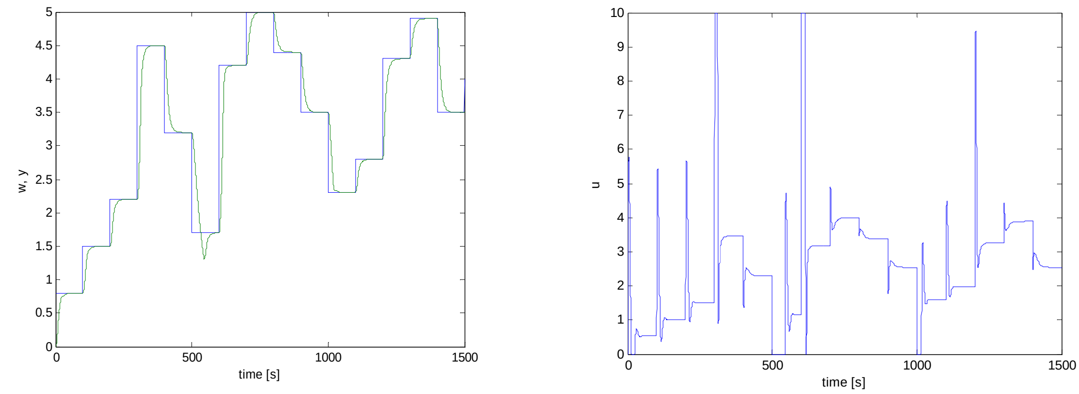

Figure 8: The Time responses of reference w, controlled y and control u variables for B system

5 Conclusion

The aim was to improve the control performance using a genetic algorithm for optimal PID controller tuning. Excellent approximately properties of MLP network was demonstrated in modelling of nonlinear controlled process. The tuning new controller parameters using genetic algorithm was succeeding very good results. The proposed controller parameter settings appear feasible and effective. Testing of quality control process was realized in simulation environment of Matlab Simulink on two types of nonlinear dynamic processes.

Acknowledgement

The work has been supported by the grants agency VEGA no. 1/0544/09 and no. 1/0690/09.

References

[1] R.C. Dorf. *Modern Control Systems*. Addison-Wesley publishing Company, 5th edition, 1990
[2] D.E.Goldberg. *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addisson-Wesley, 1989
[3] M.T.Hagan, M.B.Menhaj,. *Training Feedforward Networks with the Marquardt Algorithm.* Submitted to the IEEE Proceedings on Neural Ne, 199409
[4] A.Jadlovská. *Modelovanie a riadenie dynamických procesov s využitím neurónových sietí.* Edícia vedeckých spisov FEI TU Košice, ISBN 80-8894122-9, 2003
[5] S. Kajan, I. Sekaj. *Parameter tuning of steam pressure controller using genetic algorithm and neural model*, In: Process Control 2007 : 16th International Conference. Štrbské Pleso, Slovak Republic, 11.-14.6.2007
[6] K.F. Man, K.S. Tang, S. Kwong. *Genetic Algorithms, Concepts and Design*. Springer, 2001
[7] I.Sekaj. *Evolučné výpočty a ich využitie v praxi*, Iris 2005, Bratilava (in slovak)
[8] I.Sekaj. *Genetic Algorithm Based Controller Design*, In: 2nd IFAC conference Control System Design'03, Bratislava
[9] The Mathworks. Neural Network Toolbox,User’s Guide, documentation on CD

Ing. Slavomír Kajan, PhD, E-mail: slavomir.kajan@stuba.sk
Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava

Ing. Mária Hypiusová, PhD, E-mail: maria.hypiusova@stuba.sk
Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava