2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Application of genetic algorithms to neural networks based control of a liquid level tank system

K. Vassiljeva, and J. Belikov, and E. Petlenkov

***Abstract*— In this paper, a design of a controller based on the NN-SANARX (Neural Network based Simplified Additive Autoregressive eXogenous) model is considered on the basis of a prototype of a real liquid level tank system. Structure of the neural network is chosen using two different methods of genetic algorithms with multi-objective optimization. The goal of the control algorithm is to track the desired level of the liquid in the upper tank.**

I. Introduction

The main contribution of this paper is devoted to application of the non-analytical control method to the liquid level control problem. Design of the controller is based on the parameters of neural network of specific structure. Thus, "*black-box*" or the "*gray-box*" approach could be implemented, where some parameters of the process may be known.

While control of the liquid level is very well know problem, it is still urgent issue at the present days. In many cases the problem is managed by means of PI [1], PID [2], and fractional-order PID [3] controllers. Despite its popularity, the main drawback of PI control is that it does not by it self guarantee the same level of control accuracy on the whole operating range. Finally, methods based on computational intelligence have started to gain popularity [4].

Techniques proposed in this paper allow to automate the design of the controller and implement it in the form of software to a control laboratory plant. The main idea of the control algorithm is based on the input-output feedback linearization.

The rest of this paper is organized as follows. Section II derives description of the neural network the parameters of which are used in control design. Section III introduces two genetic algorithms used to solve the problem of multi-objective optimization. Further, implemented control technique is described. Next section gives overview of the controlled plant parameters. The simulation results are presented in Section VI to validate theoretical results. Finally, Section VII concludes the paper.

II. Preliminaries

Hereinafter, if $\xi : \mathbb{Z} \to \mathbb{R}$ and $k \in \mathbb{N}$, then $\xi^{[k]}$ stands for k-th step forward time shift of ξ and is defined by $\xi^{[k]} := \xi(t+k)$. The same is for backward shift. Moreover, to make the presentation of the material more intuitive, we restrict our attention to the case of SISO systems.

Nonlinear control systems can be represented by the discrete-time Nonlinear AutoRegressive eXogenous (NARX) models. In fact, model can be derived as governing physical laws described by the Newton equations, as well one can start from the measured i/o data of the process. In the letter case mathematical relation between variables can be found without going into the details what is actually happening inside the system. In other words new value of the output signal $y^{[n]}$ is based on the m previous values of independent input signal and its own $n-1$ previous values, i.e.

$$y^{[n]} = \varphi\left(y^{[n-1]}, \ldots, y, u^{[m]}, \ldots, u\right). \tag{1}$$

Thus, usage of neural network approach is an obvious choice.

Mathematically, neural network representation of the NARX system can be given as follows

$$y^{[n]} = \sum_{i=1}^{l} c_i \phi\Big(w_{i,1}y + \cdots + w_{i,n}y^{[n-1]} + \\ + w_{i,n+1}u + \cdots + w_{i,n+m}u^{[m]}\Big), \tag{2}$$

where $u \in \mathbb{R}$ is a real-valued scalar input, $y \in \mathbb{R}$ is a real-valued scalar output, $\phi(\cdot)$ is a saturation-type smooth nonlinear function, l is the number of hidden neurons and c_i, w_i are synaptic weights.

Usually we need a model of the process for the controller design. In order to identify the plant and obtain the state-space representation, one can use different methods. One of the possibilities is given by a subclass of NN-NARX models—Neural Network based Additive Autoregressive eXogenous models (NN-ANARX) [5]. This class of networks has a restricted connectivity: a hidden layer consists of n parallel sublayers corresponding to the nth order of the system, see [6], [7]. Such representation gives us flexibility to change easily the order of the model during identification, and write down the state-space model without additional calculations. In addition, such structure of the model guarantees linearizability via dynamic output feedback [8]. Mathematically, neural network representation of ANARX model can be formulated as follows:

$$y^{[n]} = \sum_{i=1}^{n} C_i \cdot \phi_i\left(W_i \cdot [y^{[n-i]} \;\; u^{[n-i]}]^{\mathrm{T}}\right), \tag{3}$$

where $\phi_i(\cdot)$ is an activation function of the ith sublayer neurons, C_i and W_i are $1 \times l_i$ and $l_i \times 2$ dimensional matrices of the ith sublayer output and input synaptic weights, respectively. The number of neurons on hidden layer is l_i.

K. Vassiljeva and E. Petlenkov are with the Department of Computer Control, Tallinn University of Technology (email: {kristina.vassiljeva, eduard.petlenkov}@ttu.ee) and J. Belikov is with Institute of Cybernetics, Tallinn University of Technology (email: jbelikov@cc.ioc.ee).

This work was supported by the Tiger University Program of the Information Technology Foundation for Education.

A schematic diagram of neural networks based representation of 3 can be seen in Fig. 5.

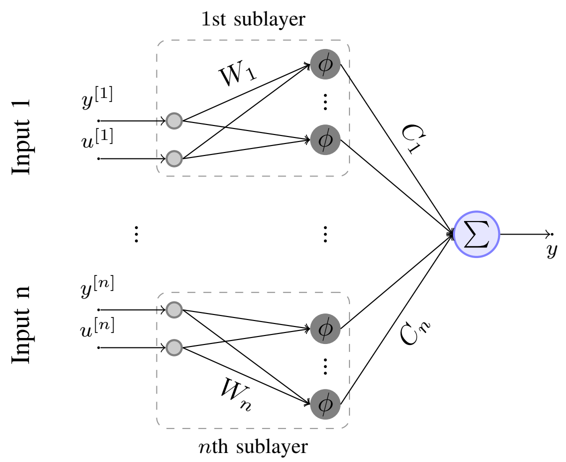

Fig. 1. Representation of NN-ANARX structure

On the other hand, while choosing the model structure one should consider not only the factors affecting the quality of identification, but also the conditions imposed by the further usage of identified model. Some applications do not require a very precise model, but the number of the parameters involved can be crucial. Therefore, the reduction of the model parameters is preferable. In that case neural network structure must undergo further transformations. Relying on available information about the regressors of the system, all redundant interconnections between hidden layers and output should be eliminated [9]. In this case Eq. (3) can be transformed to the form

$$y^{[k]} = \sum_{i=1}^{\max(n_p)} C_i \cdot \phi_i\Big(W_i \cdot [\{y_{d_{yi}}^{[i-1]}\}_{d_{yi}\in D_{yi}} \{u_{d_{ui}}^{[i-1]}\}_{d_{ui}\in D_{ui}}]^T\Big) \quad (4)$$

where

$\max(n_p)$ - maximal order considered in application;
d_{yi} - index of the previous output y on the ith layer;
D_{yi} - set of indexes d_{yi};
d_{ui} - index of the previous input u on the ith layer;
D_{ui} - set of indexes d_{ui};
and $i = 1, \ldots, m$.

It is known that relying on using NN-ANARX structure we can design a controller based on the reduced number of the parameters of the neural network using feedback linearization. The issue here is how to select the optimal structure. As it was mentioned earlier at least two objectives should be taken into account: accuracy of the control and the number of the parameters. A proven approach for the multiple-objective optimization problems are Genetic Algorithms (GA) which will be considered in the next section.

III. Multiple-objective optimization

Being a population-based approach, GA suits well to solve multi-criteria problems. For multiple-objective problems the criteria are generally conflicting, preventing simultaneous optimization of each objective. As it can be seen in our case minimization of the model parameters can decrease the control performance and vice versa.

In general, there are two different approaches to GA based optimization. One of them is to join the criteria into one composite function, for example, the *weighted sum method*. In that case the full attention should be drawn to the proper selection of the weights to characterize the decision maker preferences. On one hand that requires the level of expertise for accurate weight selection. On the other hand, it gives more flexibility to choose what is preferable in the current control application. In addition to the above mentioned scaling among objectives is needed.

The second approach provides the entire Pareto optimal set of solutions which are non-dominated with respect to each other. In other words, solution cannot be improved with respect to any objective without worsening at least one other objective [10].

In order to design a controller both techniques were taken into consideration.

A. Weighted sum approach

This method is a classical technique to solve a multi-objective optimization problem. First, we need to assign a weight k_i to each normalized objective function z_i.

$$\min z = \sum_{i=1} k_i \cdot z_i, \quad (5)$$

where $\sum k_i = 1$ and z_i is normalized objective function.

In the context of this framework those functions are

$$z_1 = |1 - e^{-\mathrm{mae}\cdot c}|, \quad (6)$$

where mae is a mean absolute error of the control, c is a scaling coefficient. Normalized order function

$$z_2 = 1 - \sum_{i=1}^{\max(n_p)} \frac{i}{\max n_p}. \quad (7)$$

As it can be seen from Eq. (5), one can find a single solution for specific k_i. However, if multiple solutions are desired, the problem should be solved several times with different combinations of weights. In this case we face choices—either rely on the level of expertise of a decision maker or use some extended technique called MOGA (Multi-Objective GA). Hence, normalized weight vector k_i is randomly generated for each solution. The main drawback of this approach is a prolongation of the chromosome, which in turn leads to slow convergence.

Based on the above discussion we came to the conclusion that in our specific case with small number of the criteria, and clear understanding of the control application purpose oriented toward the practical applications, preferences of the decision-maker should be applied. Moreover, as single objective is used to assign the fitness function, GA algorithm can be implemented with minimum modifications. In this case, general description of the procedure is not presented and could be found in [10], [11], [12], [13].

B. NSGA-II approach

The previous technique has difficulty in finding solutions uniformly distributed over a non-convex trade-off surface. One of the possible ways to overcome this obstacle is to use crowding distance approaches. The main idea is to achieve a uniform spread of solutions among best-known Pareto front, see Fig. 2. It means that fitness sharing parameter will not be used. Thus, among many variations of multi-objective GA Fast Non-dominated Sorting Genetic Algorithm (NSGA-II) was chosen. Additional points in favor of this method are: usage of the single parameter N and its efficiency [10] and [14]. The complete procedure of the NSGA-II technique is given below

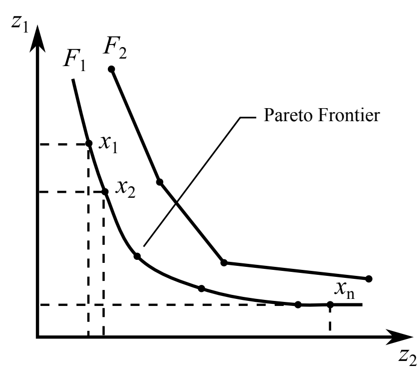

Fig. 2. Pareto Optimal set with two conflicting objectives

Step 1 : Create a random initial population P_0 of size N. Set $t = 0$.
Step 2 : Apply crossover and mutation to P_0 to create offspring population Q_0 of size N.
Step 3 : Stop and return to P_t, if stopping criterion is satisfied.
Step 4 : Set $R_t = P_t \cup Q_t$.
Step 5 : Identify and sort the non-dominated fronts $F_1, F_2, \dots F_k \in R_t$.
Step 6 : For $i = 1, \dots, k$ do the following:

a) For each objective function sort the solutions in F_i in the ascending order. Let $l = |F_i|$. Assign $cd_k(x_{[1,k]}) = \infty$ and $cd_k(x_{[l,k]}) = \infty$, and for $i = 2, \dots, l-1$

$$cd_k(x_{[i,k]}) = \frac{z_k(x_{[i+1,k]}) - z_k(x_{[i-1,k]})}{z_k^{\max} - z_k^{\min}}. \tag{8}$$

b) Total crowding distance $cd(x)$ of a solution x is the $cd(x) = \sum_k cd_k(x)$.

c) Create P_{t+1}.

Case 1: If $|P_{t+1}| + |F_i| \leq N$, then $P_{t+1} = P_{t+1} \cup F_i$;
Case 2: If $|P_{t+1}| + |F_i| > N$, then least crowded solutions from F_i to P_{t+1}.

Step 7 : Use binary tournament selection to select parents from P_{t+1}. Apply crossover and mutation to P_{t+1} to create offspring Q_{t+1}.
Step 8 : Set $t := t + 1$, goto Step 3.

IV. Control technique

If during the simulations obtained model meets the control requirements, its parameters can be used in control of the plant. As it was mentioned earlier in Section II ANARX representation guaranties linearizability via output feedback.

Dynamic output feedback can be written by using parameters of the neural network as [15]

$$\eta_1 = C_1 \phi_1 \left(W_1 \begin{bmatrix} y & u \end{bmatrix}^{\mathrm{T}} \right) \tag{9}$$

and

$$\begin{aligned} \eta_1^{[1]} &= \eta_2 - C_2 \phi_2 \left(W_2 \begin{bmatrix} y & u \end{bmatrix}^{\mathrm{T}} \right) \\ &\vdots \\ \eta_{n-2}^{[1]} &= \eta_{n-1} - C_{n-1} \phi_{n-1} \left(W_{n-1} \begin{bmatrix} y & u \end{bmatrix}^{\mathrm{T}} \right) \\ \eta_{n-1}^{[1]} &= v - C_n \phi_n \left(W_n \begin{bmatrix} y & u \end{bmatrix}^{\mathrm{T}} \right). \end{aligned} \tag{10}$$

Here $v \in \mathbb{R}$ is a reference signal (desired output). As it can be seen, the application of the dynamic feedback (9) and (10) to the model (3) or (4) results in the closed-loop system that can be described by the linear model $y^{[n]} = v$.

Further, control signal u from (10) should be extracted. There are several ways how to do that, but the simplest is to assume like in [16] that $\phi_1(\cdot)$ is a linear function. It follows from the requirements of the control algorithm (9)-(10) that the model has to be at least of the second order. Moreover, subsequent sublayer of the neural network should have nonlinear activation function in order to capture the nonlinear behavior of the plant. Thus, dynamics of the controller can be described as follows. First, define the following matrix T

$$T := C_1 \cdot W_1. \tag{11}$$

Next, it can be divided into 2 parts as

$$C_1 \cdot W_1 \cdot z_1 = T \cdot z = T_1 \cdot y + T_2 \cdot u. \tag{12}$$

Finally, control signal takes the form

$$\begin{aligned} u &= T_2^{-1} \Big(\eta_1 - T_1 y \Big) \\ \eta_1^{[1]} &= v - C_2 \phi_2 \left(W_2 \begin{bmatrix} y & u \end{bmatrix}^{\mathrm{T}} \right). \end{aligned} \tag{13}$$

Note that in the case of SISO systems $T \in \mathbb{R}^2$ is a 2×1 vector and, as a result, $T_1, T_2 \in \mathbb{R}$ are real numbers. On that basis we can arrive to conclusion that described technique can be applied only if $T_2 \neq 0$.

Fig. 3. Model of the Multi Tank system

V. Description of the Plant

A model of the Multi Tank system in Fig. 3 was provided by INTECO [17]. In the continuous time domain the dynamical behavior of the plant can be expressed by the following differential equation

$$\dot{x}_1 = \frac{1}{aw}\left(u - C_1 x_1^{\alpha_1}\right) \tag{14}$$

where $u = q$, $x_1 = H_1$. Additionally, x_1 and u are limited by the physical parameters of the system and the power of the plant. Moreover, dead zone of the control signal can be an issue that should not be neglected.

TABLE I

Nomenclature

Parameter	Physical description
H_i	fluid level in the ith tank
w	width of a tank
a	length of the upper tank
C_i	resistance of the output orifice of the ith tank
α_i	flow coefficient for the ith tank

It can be seen from Fig. 3 that all tanks are equipped with valves and piezoresistive level sensors. Also system is provided with 12 V DC pump which delivers liquid from the reservoir to the upper tank. Thus, inflow of the first tank can be controlled by the power of the pump with PWM control signal. Valves between tanks can be controlled, and therefore, change the outflows of the upper tanks and/or considered as the inputs for the lower tanks. Thus, this particular laboratory plant is reconfigurable according to the requirements. Given plant is designed to operate with an external PC-based controller, which communicates with valves, motor and sensor via RT-DAC I/O internal PCI cart. The I/O board itself is controlled by the real-time software which runs in Simulink using MATLAB Real-Time Windows Target environment.

It should be mentioned that in this particular paper, we focus on the control of the water level in the upper tank. From the above mentioned it can be easily found that in the current case we are dealing with SISO system. Physical parameters of the laboratory plant are given next $w = 0.035$ m, $a = 0.25$ m, $\alpha_1 = 0.2497$, and the maximal inflow provided by the pump is $1.0284 \cdot 10^{-4}$ m^3/s. For a full picture a resistance of the output orifice of the first tank should be known. It was determined experimentally as $C_1 = 11.08 \cdot 10^{-5}$ m^2/s, using MATLAB routine provided with the installation package.

VI. Simulation Results

The simulation results obtained applying the proposed control technique to the laboratory plant available in the Department of Computer Control, Tallinn University of Technology [18]. Parameters and structure of the neural network were found off-line a priori.

From the description of the object, as the task is to control the level of the liquid in the first (upper) tank, the order of the model should be two. Thus, the usage of the genetic algorithm is not strictly necessary in the context of the structure selection problem. Moreover, manual (trial-and-error) selection of the neural network structure is obvious choice in the particular case. However, that situation allows us to test completely the proposed controller design algorithm on a plant. So, further implementation to the more complex processes where manual selection of the structure computationally ineffective.

First, as we deal with destructive method, the maximal possible order of the model n_p should be proposed, see (4). So, in order to test efficiency of the algorithm but not waste the time, $n_p = 5$ was chosen. That leads to the fact that neural network structure representing chromosome length [19] is

$$l = n_p \cdot (n + m) \cdot m = 5 \cdot (1 + 1) \cdot 1 = 10, \tag{15}$$

where l is the length of the individual, n and m are numbers of inputs and outputs of the system, respectively. Thus, for a fully connected NN-ANARX chromosome will be as follows

$$chr = [1\ 1 \quad 1\ 1 \quad 1\ 1 \quad 1\ 1 \quad 1\ 1]. \tag{16}$$

If connection between input and output is absent, then 1 has to be replaced by 0. Continuing the above mentioned if both input and output at the same time instance are eliminated, for example $u^{[n-1]}$ and $y^{[n-1]}$, then order of the model decreases. Chromosome, representing it is $chr = [1\ 1 \quad 1\ 1 \quad 1\ 0 \quad 0\ 1 \quad 0\ 0]$, and the order is $n_p = 4$.

When decoding method is known, the next stages of the controller design procedure are described in Fig. 4. For more detailed description of finding neural networks weights and initial GA parameters see [20], [21].

Sigmoidal tangent function was taken as activation function of the sublayers, excluding the first one which is linear. Numbers of the neurons are $l_1 = 2$ and $l_i = 4$, for $i = 2, \ldots, n_p$. To perform a training the Levenberg-Marquardt (LM) algorithm was used. Both GA methods were applied for the neural network structure selection. For both cases number of the chromosomes in population is specified. This parameter remains the same during the whole process. The same individual can represent neural networks with the same structure but different weights. In other words, if performance of the controller is good, and we would like to choose that representative for future selection and/or crossover in order to form the next generation both chromosome and the neural network should be saved. Further each GA algorithm has its own preferences.

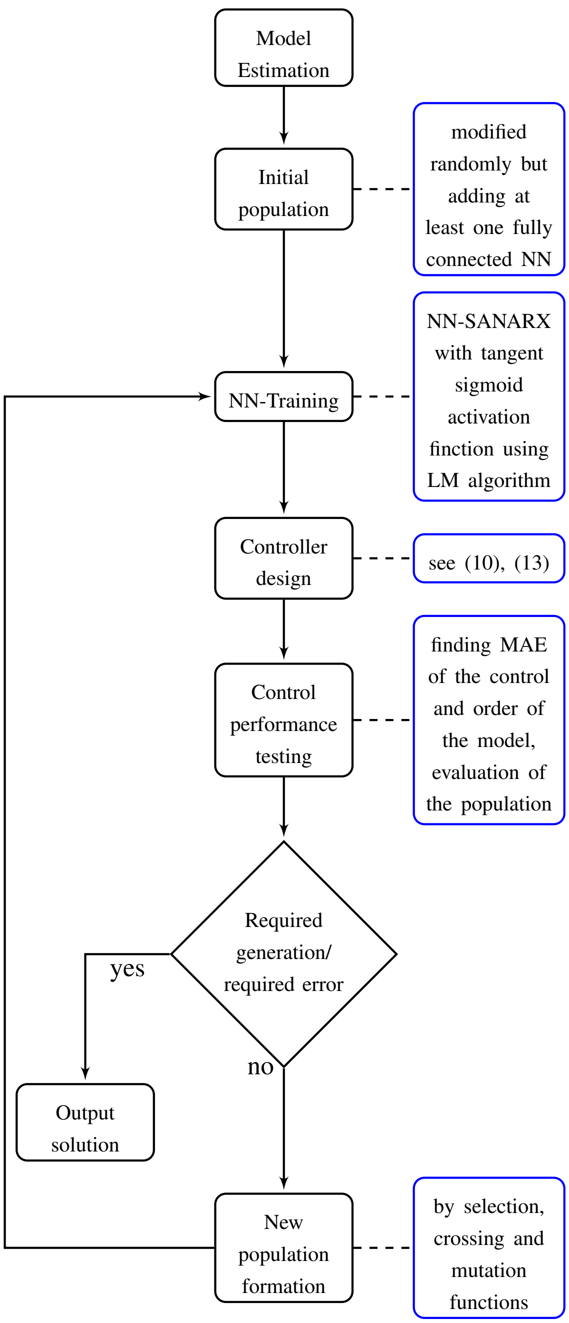

Fig. 4. Controller design conceptual algorithm

For the *weighted sum approach* percentage of crossover rate equal to 70% was chosen, mutation rate - 2%, additionally to avoid the premature convergence of the algorithm 1% of the new blood (randomly generated individual) were used. In order not to lose a good neural network at least 2 parents from the previous generation were written directly to the next one. Several combinations of weights of objective functions were tried, but the most suitable solutions were provided as follows

$$\min z = 0.45 \cdot z_1 + 0.55 \cdot z_2.$$

The scaling coefficient in the first objective function (6) $c = 10$. After 25 generations the next chromosome were obtained $chr = [1\ 1 \quad 1\ 0 \quad 0\ 0 \quad 0\ 0 \quad 0\ 0]$. According to the previously described decoding this individual represents the next structure of the neural network, see Fig. 5.

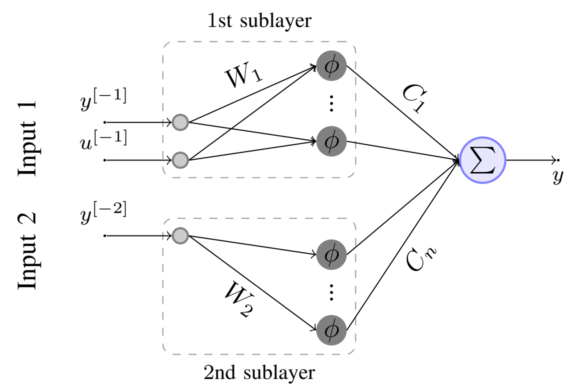

Fig. 5. Structure found with *weighted sum approach*

Results of the control are shown in Fig. 6. As it can be seen designed controller a capable of reference signal tracking.

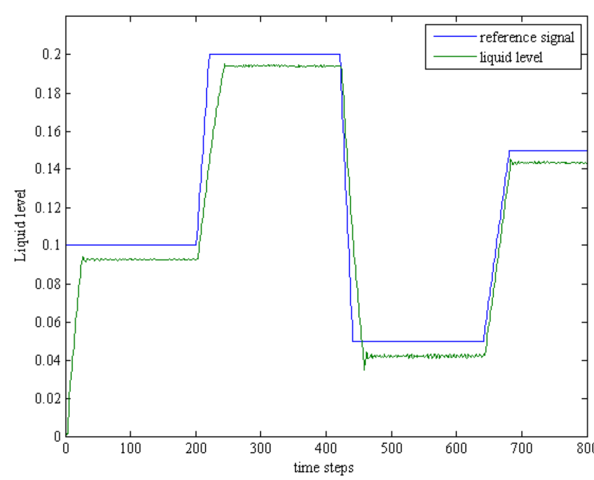

Fig. 6. Control with weighted sum method

Second GA uses a tournament selection. *NSGA-II* combines previous population with offspring population, so elitism strategy is also maintained here. After 25 generations Pareto frontier was found. As this approach gives multiple solutions, then the main problem is to find the optimal one among them. This could be a problem if Pareto frontier is large enough and solutions with $cd = \infty$ are not best ones. First ten representatives of the Pareto front were studied in the final generation. The best control performance was presented by the fifth neural network with chromosome $chr = [1\,1 \quad 1\,1 \quad 0\,0 \quad 0\,0 \quad 0\,0]$. In that case output of the system depends on both inputs at time instance $t = n - 2$.

Control performance is shown in Fig. 7. Particular de-

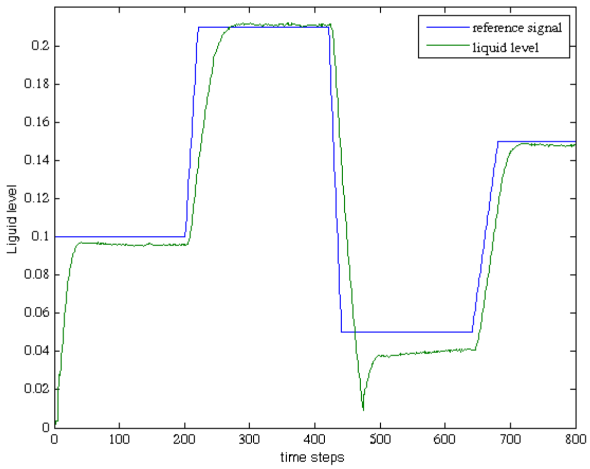

Fig. 7. Control with NSGA-II method

signed controller showed a little bit better performance than the previous one, but some overregulation and offset were obtained then reference signal decreased drastically.

VII. Conclusions

Experiments showed that both methods of neural network structure selection for controller design have good control performance of a liquid level tank system. It should be noted that initially *mse* function was used as one of the objectives. Previously carried out computer simulations showed satisfactory results. Unfortunately, in practice oscillations were increased by the dead zone of the motor. So, in order to avoid oscillations of the output signal, *mae* criterion was used. Moreover, if during simulation offset was larger than $\sigma > 10\%$, chromosome representing that neural network was given the maximal value of the $z_1 = 50$ function, and thus excluded the possibility to produce offsprings.

If number of objective functions is not large and level of expertise of decision maker is high enough then *weighed sum approach* could be preferable. Otherwise, with larger number of the criteria choice of the objective function weights becomes more complex, therefore NSGA-II approach could be used.

References

[1] K. Sundaravadivu, B. Arun, and K. Saravanan, "Design of fractional order pid controller for liquid level control of spherical tank." in *ICCSCE*, 2011, pp. 291–295.

[2] G. Yao and a. X. C. Fengxiang Gao, and Changsong Wang, "Design and simulation based on kalman filter fuzzy adaptive pid control for mold liquid level control system," in *Chinese Control and Decision Conference*, Guilin, China, 17-19 June 2009, pp. 6105 – 6109.

[3] A. Tepljakov, E. Petlenkov, J. Belikov, and M. Halas, "Design and implementation of fractional-order pid controllers for a fluid tank system," in *American Control Conference (ACC)*, Washington, DC, 17-19 June 2013, pp. 1777 – 1782.

[4] L. Liang, "The application of fuzzy pid controller in coupled-tank liquid-level control system," in *International Conference on Electronics, Communications and Control*, Ningbo, China, 9-11 September 2011, pp. 2894 – 2897.

[5] Ü. Kotta, S. Nõmm, and F. Chowdhury, "On a new type of neural network-based input-output model: The ANARMA structure," in *The 5th IFAC Symposium on nonlinear control systems NOLCOS*, St. Petersbourg, Russia, July 2001, pp. 1535–1538.

[6] Ü. Kotta, F. Chowdhury, and S. Nõmm, "On realizability of neural networks-based input-output models in the classical state space form," *Automatica*, vol. 42, no. 7, pp. 1211–1216, 2006.

[7] E. Petlenkov, S. Nõmm, and Ü. Kotta, "Nn-based anarx structure for identification and model-based control," in *The 9th International Conference on Control Automation Robotics and Vision*, Singapore, December 2006, pp. 2284–2288.

[8] E. Aranda-Bricaire, Ü. Kotta, and C. H. Moog, "Linearization of discrete-time systems," *SIAM Journal of Control and Optimization*, vol. 34, no. 6, pp. 1999–2023, 1996.

[9] K. Vassiljeva, J. Belikov, and E. Petlenkov, "Neural network based minimal or reduced model representation for control of nonlinear MIMO systems," in *Proceedings of International Joint Conference on Neural Networks*, San Jose, CA, USA, July 31 - August 5 2011, pp. 1706–1713.

[10] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering and Safety, Elsevier*, no. 91, pp. 992–1007, 2006.

[11] S. N. Sivanandam and S. Deepa, *Introduction to Genetic Algorithms*. Berlin, Heidelberg: Springer, 2010.

[12] K. Vassiljeva, J. Belikov, and E. Petlenkov, "Genetic algorithm based structure identification for feedback control of nonlinear mimo systems," in *Adaptive and Intelligent Systems: Second International Conference, ICAIS 2011, Klagenfurt, Austria, September 6-8, 2011, Proceedings*, A. Bouchachia, Ed. Springer, Lecture Notes in Artificial Intelligence, 2011, pp. 215–226.

[13] D. Whitley, "Genetic algorithms and neural networks," in *Genetic Algorithms in Engineering and Computer Science*. John Wiley, 1995, pp. 191–201.

[14] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary Computation, MIT Press*, vol. 8, no. 2, pp. 173–195, 2000.

[15] E. Petlenkov, "Model reference control of nonlinear systems by dynamic output feedback linearization of neural network based anarx models," in *The 10th International Conference on Control Automation Robotics and Vision*, Hanoi, Vietnam, 17-20 December 2008, pp. 1119–1123.

[16] E. Petlenkov, "NN-ANARX structure based dynamic output feedback linearization for control of nonlinear MIMO systems," in *The 15th Mediterranean Conference on Control and Automation*, Athena, Greece, June 2007, pp. 1–6.

[17] INTECO, AGH University of Science and Technology. (2013) Inteco. [Online]. Available: http://www.inteco.com.pl

[18] Department of Compuer Control at Tallinn University of Technology. (2013) Alpha control laboratory. [Online]. Available: http://a-lab.ee

[19] K. Vassiljeva, E. Petlenkov, and J. Belikov, "Neural network based minimal state-space representation of nonlinear MIMO systems for feedback control," in *2010 the 11th International Conference on Control Automation, Robotics and Vision*, Singapore, 7-10 December 2010, pp. 2191–2196.

[20] J. Belikov, E. Petlenkov, S. Nõmm, and K. Vassiljeva, "Computational intelligence methods based design of closed-loop system," in *Neural Information Processing : 20th International Conference, ICONIP 2013, Daegu, Korea, November 3-7, 2013, Proceedings, Part I*, M. Lee, A. Hirose, Z.-G. Hou, and R. Kil, Eds. Springer, Lecture Notes in Computer Science, 8226, pp. 215 – 224.

[21] K. Vassiljeva, E. Petlenkov, and S. Nõmm, "Evolutionary design of the closed loop control on the basis of nn-anarx model using genetic algorithm," in *Neural Information Processing : 19th International Conference, ICONIP 2012, Doha, Qatar, November 12-15, 2012, Proceedings, Part I*, T. Huang, Z. Zeng, C. Li, and C. S. Leung, Eds. Springer, Lecture Notes in Computer Science, 7663, pp. 592 – 599.