HYBRID PID CONTROL ALGORITHMS FOR NONLINEAR PROCESS CONTROL

Albena Taneva — Michail Petrov — Ivan Ganchev *

This paper presents modifications of the classical PID control algorithm, implemented by an Adaptive Neuro-Fuzzy Architecture (ANFA). The main goal here is to design a fuzzy PID controller with a flexible structure, adaptive tuning of its parameters and algorithm modifications, which leads to improvement of the system performance. Thus the controlling process and system are prevented from the undesired and non expected changes of the system input signals. The antecedent part of the applied fuzzy rules contains a linear function, similar to the modified discrete equation of the corresponding conventional PID controller. The simulations demonstrate satisfactory results of these performances and implementations applied to a nonlinear plant.

Keywords: adaptive control algorithm, hybrid fuzzy PID

1 INTRODUCTION

Fuzzy logic controller (FLC) has emerged as one of the most active and useful research areas in fuzzy control theory. Therefore fuzzy logic controllers have been successfully applied in the control of various physical processes. On the other hand the best-known industrial process controller is the Proportional-Integral-Derivative (PID) controller because of its simple structure and robust performance in a wide range of operating conditions. The similarity between FLC and PID controllers and their improvement is still investigated. This paper is devoted to this problem and describes some of the design aspects of the fuzzy PID controllers with application to nonlinear plants.

Three types of the structure of the FLC have been studied: the first one is the well known fuzzy PD controller, which generates a control action (u) from the system error (e) and the change in the error (Δe), the second one is the fuzzy PI controller, which generates an incremental control action (Δe) from the error (e) and the change in the error (Δe). The fuzzy PD controller is a positioning type controller, and the fuzzy PI controller is a velocity type controller. The third one is the fuzzy PID controller, which generates a control action (u) from the error (e), the change in the error (Δe) and the sum of errors (δe) or the fuzzy PID controller, which generates an incremental control action (Δu) from the error (e), the change in the error (Δe) and the acceleration error ($\Delta^2 e$). The first type of a fuzzy PID controller is a positioning type controller and the second type is a velocity type controller. The difficulties of both types fuzzy PID controllers are that they need three inputs, which will expand the rule-base significantly and will make the design procedure more complicated. Therefore such types of fuzzy PID controllers are rarely used. The fuzzy PD and fuzzy PI controllers based on Mamdani's [1] fuzzy system are simpler and more applicable. The fuzzy PI type control is known to be more practical than PD type because it is difficult for the PD type to remove the system steady state error. On the other hand the PI type control is known to give poor performance in the system transient response for higher order process due to the internal integration operation. Therefore the main goal here is to design a PID controller with a flexible structure, adaptive tuning of its parameters and based on certain modifications, appropriate for nonlinear control systems with variable system reference input.

2 STRUCTURE OF A CONTROL SYSTEM WITH AN ANFA PID CONTROLLER

The structure of the control system with the proposed ANFA PID controller is shown in Figure.1. An additional conventional PD controller works in parallel with the main Fuzzy PID controller, because a learning method for neural network investigated by Gomi and Kawato [2] is used. They have proposed learning schemes using feedback error learning for a neural network model applied to an adaptive nonlinear feedback controller. In these learning schemes, a conventional feedback controller (CFC) is used both as an ordinary feedback controller to guarantee global asymptotic stability in a particular space and as an inverse reference model of the response of the controlled plant [11]. This approach is implemented here in order to tune the fuzzy neural network.

The traditional FLC works with input signals of the system error e and the change Δe of the error. The

* Technical University of Sofia, branch Plovdiv, 25 Tsanko Dyustabanov, 4000 Plovdiv, Bulgaria; altaneva@tu-plovdiv.bg, mpetrov@tu-plovdiv.bg, ganchev@tu-plovdiv.bg

ISSN 1335-3632

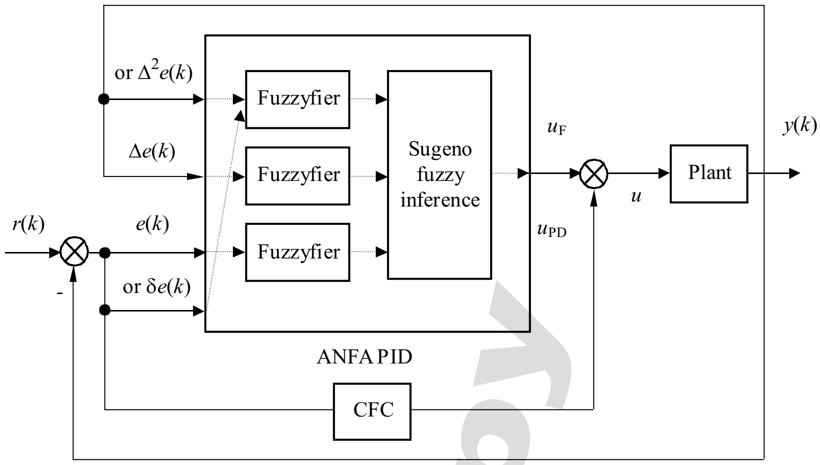

Fig. 1. The structure of the control system with the proposed modified ANFA PID controller

system error is defined as the difference between the set point $r(k)$ and the plant output $y(k)$ at the moment k

$$e(k) = r(k) - y(k) \tag{1}$$

and the change of the error Δe at the moment k for the traditional PID control algorithm is calculated as follows:

$$\Delta e = e(k) - e(k-1)\,. \tag{2}$$

For a modified PID control algorithm, where the set point could be excluded from the derivative part, calculation can be done with the system output signal

$$\Delta_m(k) = -\big(y(k) - y(k-1)\big)\,. \tag{2a}$$

The sum of the errors δ_e or the acceleration error $\Delta^2 e$ can be used as a third input signal for the ANFA PID. They are calculated according to the equations:

$$\delta e(k) = \sum_{i=1}^{k} e(i) \tag{3}$$

$$\Delta^2 e(k) = e(k) - 2e(k-1) + e(k-2)\,. \tag{4}$$

For the modified algorithm the acceleration error $\Delta^2 e_m$ is calculated as follows

$$\Delta^2 e_m(k) = -y(k) + 2y(k-1) - (k-2)\,. \tag{4a}$$

The first one (Equ. 3) is attached to the positioning type FPID controller and the second one (Equ. 4) is attached to the velocity type FPID controller as it is explained bellow. It is known from the digital control theory, that the most frequently used digital PID control algorithm can be described with the difference equations as follows [3]

- positioning type PID controller: standard and modified form

$$u(k) = k_p e(k) + k_i \delta e(k) + k_d \Delta e(k)\,, \tag{5}$$

$$u(k) = k_p e(k) + k_i \delta e(k) + k_d \Delta e_m(k)\,. \tag{5a}$$

- velocity type PID controller: standard and modified form

$$\Delta u(k) = k_p \Delta e(k) + k_i e(k) + k_d \Delta^2 e(k)\,, \tag{6}$$

$$\Delta u(k) = k_p \Delta e_m(k) + k_i e(k) + k_d \Delta^2 e_m(k)\,. \tag{6a}$$

where $k_i = k_p \frac{T_k}{T_i}$, $k_d = k_p \frac{T_k}{T_i}$, T_k is the sample time of the discrete system, T_i is the integral time constant of the conventional controller, T_d is the differential time constant, k_p is the proportional gain, $u(k)$ is the output control signal and $\Delta u(k)$ is the incremental control signal. The final control action for the controller (6, 6a) can be calculated according to the previous value of the control output $u(k-1)$ as follows

$$u(k) = u(k-1) + \Delta u(k)\,. \tag{6b}$$

The Takagi-Sugeno-Kang (TSK) fuzzy rules into the ANFA PID controller can be composed in the generalized form of if-then' composition with a premise and an antecedent part to describe the control policy. The rule base comprises a collection of N rules, where the upper index (n) represents the rule number, e, Δe, $\Delta^2 e$, δe are the input variables. The modified forms and their fuzzy-neural implementation are considered in this work. The main advantage of these forms is the absence of the so called "differential kick" and reaching"bumpless" behavior of the controller output. Their application is appropriate for systems with frequent and big changes of the system set point value $r(k)$. This way the similarity between the equations of the conventional digital PID controller (5), (6) and the Sugeno output functions f_u into (7) and (8) could be found:

- positioning type ANFA PID controller: standard and modified form

$R^{(n)}$ if e is $E_i^{(n)}$ and Δe is $\mathrm{d}R_i^{(n)}$ and δe is $\delta E_i^{(n)}$ then

$$f_u^{(n)} = k_p^{(n)} e(k) + k_i^{(n)} \delta e(k) + k_d^{(n)} \Delta e(k) + k_0^{(n)}\,, \tag{7}$$

$R^{(n)}$ if e is $E_i^{(n)}$ *and*Δe is $\mathrm{d}E_i^{(n)}$ *and*δe is $\delta E_i^{(n)}$ then

$$f_u^{(n)} = k_p^{(n)} e(k) + k_i^{(n)} \delta e(k) + k_d^{(n)} \Delta e_m(k) + k_0^{(n)}\,. \tag{7a}$$

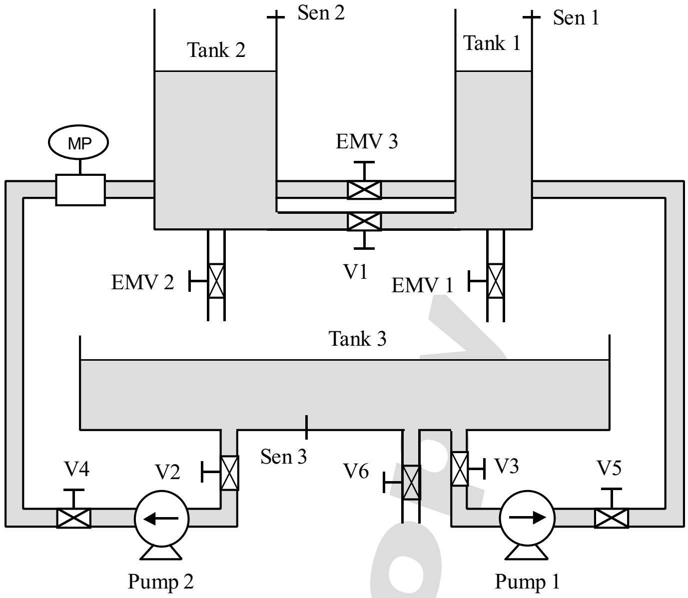

Fig. 2. Overall scheme of the plant

- velocity type ANFA PID controller: standard and modified form

$R^{(n)}$ if e is $E_i^{(n)}$ and Δe is $\mathrm{d}E_i^{(n)}$ and Δ^2 is $\mathrm{d}^2E_i^{(n)}$ then

$$f_u^{(n)} = k_p^{(n)}\Delta e(k) + k_i^{(n)} e(k) + k_d^{(n)}\Delta^2 e(k) + k_0^{(n)}, \quad (8)$$

$R^{(n)}$ if e is $E_i^{(n)}$ $and \Delta e$ is $\mathrm{d}E_i^{(n)}$ $and \Delta^2 e$ is $\Delta^2 E_i^{(n)}$ then

$$f_u^{(n)} = k_p^{(n)}\Delta e_m(k) + k_i^{(n)} e(k) + k_d^{(n)}\Delta^2 e_m(k) + k_0^{(n)}. \quad (8a)$$

In this case, the ANFA PID controller can be considered as a collection of many local PID controllers, which are represented by the (TSK) functions into the different fuzzy rules and this way can approximate the nonlinear characteristic of the controlled plant.

The fuzzy implication, connected to the rules, is realized by means of the composition [4]

$$\mu_u^{(n)} = \mu_u^{(n)} * \mu_{\Delta e}^{(n)} * \mu_{\delta e}^{(n)}\,, \quad (9)$$

$$\mu_u^{(n)} = \mu_e^{(n)} * \mu_{\Delta e}^{(n)} * \mu_{\Delta^2 e}^{(n)} \quad (10)$$

where μ_e, $\mu_{\Delta e}$, $\mu_{\delta e}$ and $\mu_{\Delta^2 e}$ specify the membership degrees upon the fired fuzzy sets of the input signals into the $(n)^{\text{th}}$ fuzzy rule. For a discrete universe with m quantization levels in the fuzzy output, the control action u_F is expressed as a weighted average of the (TSK) output functions f_u and their membership degrees μ_u of the quantization levels

$$u_F = \frac{\sum_{i=1}^{q} f_u^{(i)} \mu_u^{(i)}}{\sum_{i=1}^{q} \mu_u^{(i)}} \text{ or } u_F = \sum_{i=1}^{q} f_{ui}\overline{\mu}_{ui}\,. \quad (11)$$

For simplicity the number of the rule is represented with upper index (i).

3 THE STRUCTURE OF THE FUZZY NEURAL PID CONTROLLER

A connectionist model of the FPID controller implemented as a fuzzy neural network — ANFA is presented. The neural network structure corresponds to the fuzzy controller structure (Figure. 1) almost one to one. The input nodes in the first layer are $X1$, $X2$ and $X3$ connected to the fuzzification μ-modules in the second layer. The R-modules from third layer interpret the rules and give their output to the μ_u-modules in the fourth layer related to the control action u_F that is formed by the output U-node in the fifth layer. The nodes in layer two are term nodes, which act as membership functions to represent the terms of the respective linguistic variables. This structure enables adaptation of the controller properties according to the changing process parameters and environment.

Every node in the second layer performs a simple membership function. For example a triangular function is used in this case with expressions

$$\mu_{ji} = \begin{cases} \frac{x-a_{ij}}{b_{ji}-a_{ji}}\,, & \text{if } a_{ij} \le x \le b_{ji}\,, \\ \frac{c_{ji}-x}{c_{ji}-b_{ji}}\,, & \text{if } b_{ji} \le x \le c_{ji} \end{cases} \quad (12)$$

where x_i is the input from the i^{th} input node ($X1$, $X2$ or $X3$), μ_{ij} is the membership function of the j^{th} term of the i^{th} input linguistic variable; b_{ij}, a_{ij}, and c_{ij} are respectively the center (or mean), the left distance (or variance) and the right distance of the triangular function of the j^{th} term of the i^{th} input linguistic variable. Hence the unity link 'weight' in layer 2 μ_{ij} can represent up to three parameters – - b_{ij}, a_{ij}, c_{ij}. The parameters in this layer are adjustable and are referred to as premise parameters. The links in layer 3 are used to perform precondition matching of the fuzzy logic rules. Hence, the rule nodes $R(n)$ should perform collection of the membership degrees of the fired fuzzy sets. The links in layer 4 should perform the fuzzy product' operation according to (9) or (10) and to integrate the fired rules which have the same consequent. The single node in layer 5 computes the overall output signal as the summation of all incoming signals.

$$I_U = \sum_{i=1}^{m} f_{ui}\mu_{ui}\,. \tag{13}$$

This node transmits the decision signal out of the network and in this way acts as TSK output defuzzyfication

$$u_F = \frac{\sum_{i=1}^{m} f_{ui}\mu_{ui}}{\sum_{i=1}^{m}\mu_{ui}} \quad \text{or} \quad u_F = \sum_{i=1}^{m} f_{ui}\overline{\mu}_{ui} \tag{14}$$

where $\overline{\mu}_{ui}$ is the normalized value of μ_{ui}. The link weight μ_i presents the coefficients (k_p, k_i, k_d, k_o) into the TSK output function (7) or (8). Parameters β_i in this layer are adjustable and will be referred to as consequent parameters. The premise and the consequent adjustable parameters are taken into the learning algorithm, described bellow.

3.1. Learning algorithm RTGA for the fuzzy-neural implemented PID controller

The described fuzzy neural PID controller implements the basic control function in the system, and the conventional feedback PD controller is used for the learning algorithm (Fig. 1). The control action is obtained as a sum of the output signal from the ANFA PID controller u_F and the output signal from the conventional PD controller u_{PD}, working in parallel

$$u = u_{PD} + u_F\,. \tag{15}$$

The learning algorithm is based on instant minimization of an error measurement function, which is defined as

$$E = \varepsilon^2/2 \tag{16}$$

where ε is calculated as a difference $\varepsilon = u - u_F = u_{PD}$, in which u denotes desired control action and u_F is calculated by the neural network.

This algorithm performs two-steps gradient learning procedure — Recurrent Two-steps Gradient Algorithm — RTGA. Assuming that β_{ij} is an i^{th} adjustable parameter (*eg* the constant k_p, k_i, k_d, or k_o) in the TSK output function f_u (7) or (8) into the j^{th} activated rule, which is represented as a connection for the output neuron in the fifth layer, the general parameter learning rule used is [5, 10]

$$\beta_i(k+1) = \beta_i(k) + \eta\Big(-\frac{\partial E}{\partial \beta_i}\Big), i = 0, 1, 2, 3;\; j = 1, 2, \dots, q \tag{17}$$

where η is the learning rate, and the derivative of the error is calculated by partial derivatives.

After calculating the partial derivatives, the final recurrent equation for each adjustable parameter β_ii (k_p, k_i, k_d, k_o) in the fifth layer is:

- positioning type ANFA PID controller ((2a) is used for modified form)

$$\begin{aligned} k_p(k+1) &= k_p(k) + \eta u_{PD}\overline{\mu}_{ui}e(k)k_i(k+1) \\ &= k_i(k) + \eta u_{PD}\overline{\mu}_{ui}\delta e(k)\,, \\ k_d(k+1) &= k_d(k) + \eta u_{PD}\overline{\mu}_{ui}\Delta e(k)\,, \\ k_o(k+1) &= k_o(k) + \eta u_{PD}\overline{\mu}_{ui}\,. \end{aligned} \tag{18}$$

- velocity type ANFA PID controller ((4a) is used for modified form)):

$$\begin{aligned} k_p(k+1) &= k_p(k) + \eta u_{PD}\overline{\mu}_{ui}\Delta e(k)\,, \\ k_i(k+1) &= k_i(k) + \eta u_{PD}\overline{\mu}_{ui}e(k)\,, \\ k_d(k+1) &= k_d(k) + \eta u_{PD}\overline{\mu}_{ui}\Delta^2 e(k)\,, \\ k_o(k+1) &= k_o(k) + \eta u_{PD}\overline{\mu}_{ui}\,. \end{aligned} \tag{19}$$

The next two layers: forth and third, do not contain adjustable parameters. Therefore the output error E can be propagated back directly to the second layer, with adjustable parameters α_{ij}. The error E is propagated through the links composed by corresponding membership degrees $\mu_{ui} - \mu_{ij}$ from the fifth layer to the second layer. Hence, the learning rule for the second group of adjustable parameters in the second layer can be taken from

$$\alpha_i(k+1) = \alpha_i(k) + \eta\Big(-\frac{\partial E}{\partial \alpha_{ij}}\Big). \tag{20}$$

After calculating the partial derivatives, the final recurrent equations for each adjustable premise parameter for the i^{th} input and its j^{th} fuzzy set in the second layer is

$$\alpha_{ji}(k+1) = \alpha_{ji}(k) + \eta u_{PD}\frac{f_{ui} - u_F}{\sum_{i=1}^{m}\mu_{ui}}\frac{\partial \mu_{ji}}{\partial \alpha_{ji}}\,. \tag{21}$$

4.1 Plant description

The investigations were carried out by simulations of the Modified ANFA PID algorithms in MATLAB/Simulink environment. A simplified nonlinear model of three tanks is used. Two of them are used for level control and third one is used as a reservoir. The plant has two control inputs corresponding to the two pumps. The process variables are the levels in each tank. The model can be configured as MIMO or cascade plant model. Hence the level in a tank can be controlled through level in the other tank. Simulations were performed in cases with different references and with added disturbances in the plant.

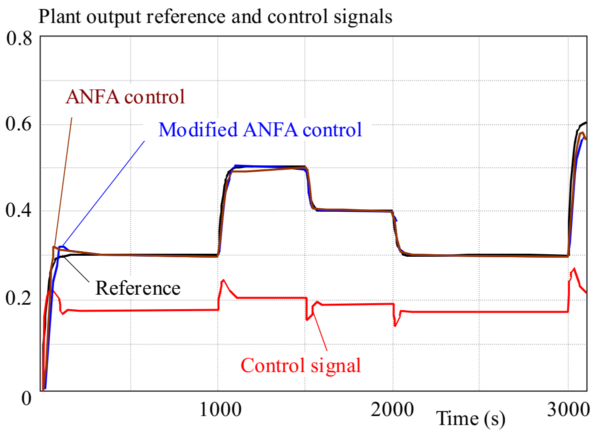

Fig. 3. Transient response with modified ANFA PID algorithm

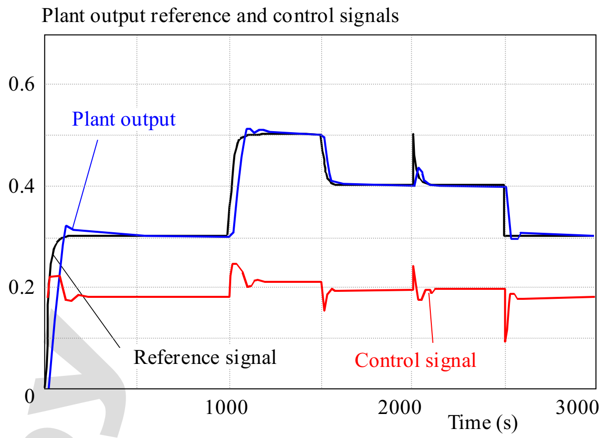

Fig. 4. Transient response and reference change

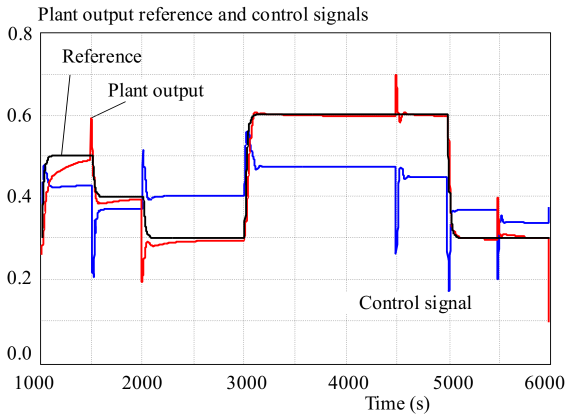

Fig. 5. Transient response to the plant output disturbance

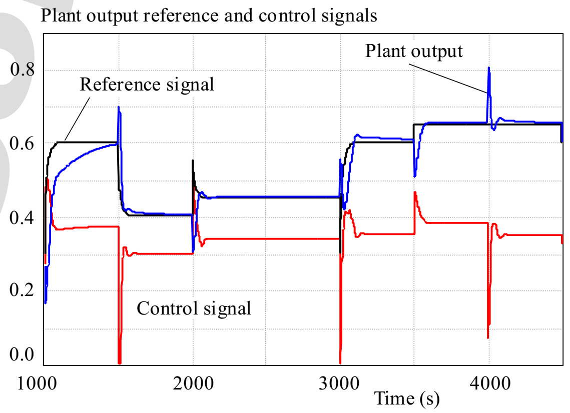

Fig. 6. Transient response to the plant output disturbance and reference change

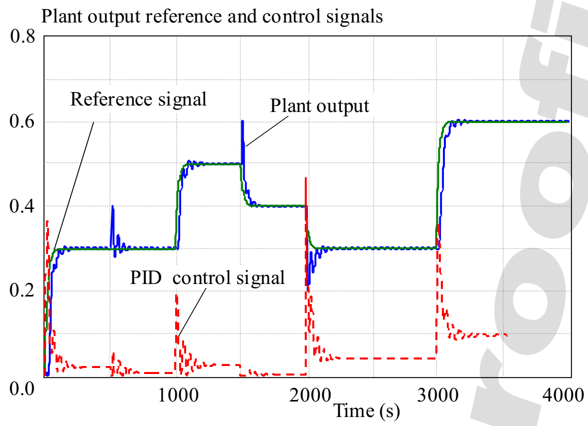

Fig. 7. Transient response with applied Conventional PID control

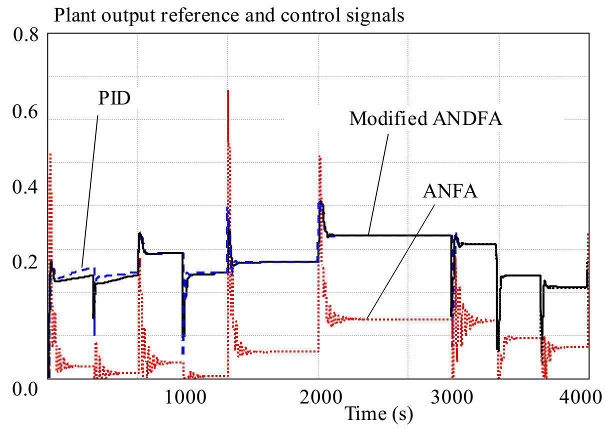

Fig. 8. Control signals of the PID, ANFA PID and Modified ANFA PID controllers

4.2 Simulation results

In this section the simulation results obtained using the developed algorithms are presented. Figure 3 shows the transient responses with ANFA and modified ANFA controllers. It was obtained different responses. The main advantage can be noted when the reference change kicks up the output signal, shown on Figure 4. With modified ANFA algorithm the controller signal remains accordingly to the common system behavior. In case of added disturbances in the system it was found out that the modified ANFA controller produces the appropriate control signal, hence the plant output closely follows the reference signal, Figure 5.

Figure 6 shows the transient responses with disturbance and with uncertain references. The obtained result are satisfied and proved the advantage of the proposed modification. Transient responses with ANFA PID and conventional PID control for the comparison and in the same cases are presented, Figure 7. The final Figure 8 shows only the control signals of the three algorithms: Modified ANFA, ANFA and conventional PID control signals. It is evident that the first one has better behavior, hence it is appropriate to be used when the reference is changed in a supervisory control system.

5 CONCLUSIONS

The fuzzy PID controllers are very useful when the controlled plant has got nonlinearities or changeable parameters. In this paper an improvement of neuro-fuzzy PID algorithms for a nonlinear plant is presented. The ANFA PID controllers are developed as three-term fuzzy controllers using the system error, the first and the second derivatives of the error (or accumulative error). The antecedent part of the applied TSK fuzzy rules contains a linear function, similar to the discrete equation of the digital PID controller. The modified forms and their fuzzy-neural implementation are considered in this work. The known modification of the conventional PID control law (about the differential part) is implemented with ANFA structure. Hence the fuzzy rules contain the modified equations as antecedent parts. The main goal was to investigate the efficiency of the modified neuro-fuzzy algorithms. The main advantage of these forms is the absence of the so called "differential kick" and reaching "bumpless" behaviour of the controller output. Their application is appropriate for systems with frequent and big changes of the system set point value. The computer simulations were carried out with Simulink model of cascaded tanks with changeable parameters. The results verified the validity and the robust performance of the system with the proposed modified fuzzy controller.

Acknowledgments

The work has been supported by the Ministry of Education and Science of Bulgaria, Research Fund Project No: BY-TH-108/2005.

References

[1] MAMDANI, E. H.: Application of Fuzzy Algorithms for Control of Simple Dynamic Plant, Proceedings IEEE **121** No. 12 (1974).

[2] GOMI, H.—KAWATO, M.: Neural Network Control for a ClosedLoop System Using FeedbackErrorLearning, Neural Networks **6** (1993).

[3] ÅSTROM, K-J.—WITENMARK, B.: Computer Controlled Systems, 3 ed., Prentice Hall, 1997.

[4] TAKAGI, H.—SUGENO, M.: FUZZY Identification of Systems and its Applications to Modeling and Control, IEEE Trans. on Systems, Man., and Cybern **SMC-15**(1) (1985), 116–132.

[5] LIN, C. T.: Neural Fuzzy Control Systems with Structure and Parameter Learning, World Scientific, NY, 1994.

[6] PETROV, M.—GANCHEV, I.—TANEVA, A.: Overview of Fuzzy Neural Control Applications, Preprints of the International Summer School on "Automation'2001", Maribor, Slovenia, 16–24 June, 2001, pp. 116–122.

[7] ZILKA, V.—HALÁS, M.—HUBA, M.: Nonlinear Controllers for a Fluid Tank System, Lecture Notes in Computer Science Vol. 5717, Computer Aided Systems Theory EUROCAST 2009, Springer-Verlag, Berlin Heidelberg, 2009, pp. 618-625.

[8] TZUU-HSENG, S. Li—MING-YUAN SHIEH: Design of a GA-Based Fuzzy PID Controller for Non-Minimumphase Systems, Fuzzy Sets and Systems **111** (2000), 183–197.

[9] SHI, Y.—MUZIMOTO, M.: Some Considerations on Convention Neuro-Fuzzy Learning Algorithms Gradient Descent Method, Fuzzy Sets and Systems **112** (2000), 51–63.

[10] REZNIK, L.—GHANAYEM, O.—BOURIMISTROV, A.: PID Plus Fuzzy Controller Structures as a Design Base for Industrial Applications, Fuzzy Sets and Systems **16** (2000), 419–480.

Received 23 March 2010

Albena Taneva
Michail Petrov
Ivan Ganchev