Decoupling Fuzzy-Neural PID Controller for TITO Nonlinear Systems

I. Ganchev, S. Ahmed, A. Taneva, M. Petrov

Keywords: Decoupling control; fuzzy PID controller; self-tuning PID controller.

Abstract. This paper presents a fuzzy-neural structure of a Decoupling Fuzzy PID controller with self-tuning parameters. This structure is appropriate for Two-Input-Two-Output (TITO) nonlinear system. The main advantage here is that the equation of classical PID control and decoupling coefficients are used as a Sugeno function into the fuzzy rules. Hence the designed decoupling fuzzy PID controller can be viewed as a natural similarity to the conventional one with decoupling elements. A benchmark quadruple tank, implementing a TITO nonlinear system is considered to illustrate the benefits of the design paradigm. The performance of this set up was studied for reference tracking and disturbance rejection cases. Simulation results confirm the effectiveness of the proposed solution.

Introduction

Fuzzy logic controllers (FLC) have emerged as one of the most active and useful research areas in fuzzy control theory. That is why they have been successfully applied for control of various physical processes. On the other hand, the best-known industrial process controller is the Proportional-Integral-Derivative (PID) because of its simple structure and robust performance in a wide range of operating conditions. Fuzzy PID (FPID) controller can be viewed as a natural similarity to the conventional PID controller for applications in nonlinear control systems.

Fuzzy and neural control, in particular, has had an impact in the control community because of their simplicity and feasibility to use heuristic control knowledge for control problems. The integration of fuzzy logic with neural network techniques has resulted in what is commonly referred to as neuro-fuzzy systems. These systems use fuzzy rules as the underlying structure and then apply neural techniques to learn the rule parameters, e.g., the input region covered by each rule and the output value of each rule.

Recently, many analysis results and design methodologies of fuzzy system and neural network have been reported. Most of the reported research however only focused on SISO systems [8-15]. The Multiple-Input Multiple-Output (MIMO) systems usually possess characteristics of nonlinear dynamics coupling [1-6]. Therefore, the difficulty of MIMO systems control is how to overcome the coupling effects between each degree of freedom [4].

In parallel, many control systems are designed in decentralized structures using a relevant number of simple single-input single-output (SISO) controllers enabling a

stable operation. Naturally, the first step of designing the decentralized control is to determine the structure and control configuration. Respectively, the decentralized control relies on control techniques applicable for SISO systems. The decoupler is to diagonalize the considered system, adding a minimal amount of dynamics and a minimal amount of time delay to the system.

There are two challenging design issues to be addressed in designing fuzzy system, viz., structural and parameter identification. The structure identification amounts to determine the proper number of rules needed i.e. finding how many rules are necessary and sufficient to properly model the available data and the number of membership functions for input and output variables. Parameter learning phase is used to tune the coefficients of each rule, like the shape and positions of membership functions. Fast computation speed is attained by requiring much less tunable parameters. There is need for effective methods for tuning the membership functions so as to minimize the output error measure or maximize performance index.

This paper is devoted to this problem and describes some of the design aspects of the fuzzy PID controllers with application to TITO nonlinear systems. Nonlinear fuzzy membership functions are utilized to implement in the network structure, called fuzzy neural network (FNN). Thus, the FNN has more design degrees of freedom to enhance the performance of the fuzzy PID controller. A simple fuzzy PID type method based on Sugeno's fuzzy technique introduced by Petrov et al. [6] is extended with decoupling control in order to improve the performance of the controller.

Another recent publication on a similar topic is published in [7]. The new simple method was presented for reducing the decoupling effect of a TITO Generalized Predictive Control (GPC) algorithm with proper adaptation of the fuzzy model and controller parameters. The advantage of fuzzy model based predictive control is proved and ensured.

This paper is organized as follows. Section 2 presents a decoupling control system structure and introduces decoupling control scheme using the adaptive FNN. The implemented design of the decoupled FNN controller and the developed on-line learning algorithm are presented in section 3. Simulation results and some concluding remarks are given in section 4 and 5 respectively.

Decoupling Control System Structure

The goal of decoupling control is to eliminate complicated loop interactions so that a change in one process

variable will not cause corresponding changes in other process variables. To do this a non-interacting or decoupling control scheme is used. In this scheme, a compensation network called a decoupler is used between the inputs of the controllers. This decoupler is the inverse of the gain array and allows for all measurements to be passed through it in order to give full decoupling of all of the loops. The structure of the decoupling control system is shown in figure 1.

Figure 1. Structure of the decoupling control system

The system errors are inputted to the conventional PD controllers according the structure proposed by Gomi and Kawato [4]. In this paper, this structure is extended by a fuzzy PID self tuning controller (FPID). As it could be seen in Petrov et. al. [6], the conventional PD controller is used both as an ordinary feedback controller to guarantee global asymptotic stability in a compact space and as an inverse reference model of the response of the controlled object.

The proposed FPID control algorithm ensures adaptive properties and deals with system disturbances and uncertainties. It combines the features of conventional PID control algorithm with intuitiveness of fuzzy logic and the adaptation abilities of artificial neural networks.

Design of Decoupled Fuzzy-Neural PID Controller

The traditional FLC works usually with input signals of the system error $e(k)$ and the change rate $\Delta e(k)$ in the error. The system error for the first and second control loop are defined as a difference between the set point $r(k)$ and the plant output $y(k)$ at the moment k , for each controller in figure 1.

$$(1) \quad e_1(k) = r_1(k) - y_1(k), \quad e_2(k) = r_2(k) - y_2(k).$$

Hence the change rate in the error $\Delta e(k)$ at the moment k will be:

$$(2) \quad \Delta e_1(k) = e_1(k) - e_1(k-1), \quad \Delta e_2(k) = e_2(k) - e_2(k-1).$$

Each FPID controller can use as a third input signal the sum of the system errors $\delta e(k)$ or the acceleration error $\Delta^2 e(k)$, which are calculated using equations [1]:

$$(3) \quad \delta e_{1,2}(k) = \delta e_{1,2}(k-1) + e_{1,2}(k);$$

$$(4) \quad \Delta e_{1,2}(k) = e_{1,2}(k) - 2e_{1,2}(k-1) + e_{1,2}(k-2).$$

The third input ((3) or (4)) of the proposed controller depends on the selected structure for the PID controller.

It is known from discrete control theory, that the most frequently used discrete PID control algorithm can be described with the well-known equations as follows [1]:

positioning PID controller:

$$(5) \quad u(k) = k_p e(k) + k_i \delta e(k) + k_d \Delta e(k)$$

incremental PID controller:

$$(6) \quad \Delta u(k) = k_p \Delta e(k) + k_i e(k) + k_d \Delta^2 e(k)$$

where $k_i = k_p T_k / T_i$, $k_d = k_p T_d / T_k$, 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 action and $\Delta u(k)$ is the incremental control action.

The final control action for implementation of the second controller (6) can be calculated according to the previous value of the control output $u(k-1)$ as follow:

$$(7) \quad u(k) = u(k-1) + \Delta u(k).$$

The Sugeno's fuzzy rules (Takagi and Sugeno, [5]) into the FPID 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 (8) and (9), where the upper index (r) represents the rule number:

First FPID controller:

$$(8) \quad \begin{aligned} R_1^{(r)} : & \text{if } e_1 \text{ is } \tilde{E}_1 \text{ and } \Delta e_1 \text{ is } \tilde{\Delta E}_1 \text{ and } \delta e_1 \text{ is } \tilde{\delta E}_1 \text{ then} \\ f_{u1}^{(r)} &= k_{1p}^{(r)} e_1(k) + k_{1i}^{(r)} \delta e_1(k) + k_{1d}^{(r)} \Delta e_1(k) + k_{21}^{(r)} e_2(k) + k_{1o}^{(r)} \end{aligned}$$

Second FPID controller:

$$(9) \quad \begin{aligned} R_2^{(r)} : & \text{if } e_2 \text{ is } \tilde{E}_2 \text{ and } \Delta e_2 \text{ is } \tilde{\Delta E}_2 \text{ and } \delta e_2 \text{ is } \tilde{\delta E}_2 \text{ then} \\ f_{u2}^{(r)} &= k_{2p}^{(r)} e_2(k) + k_{2i}^{(r)} \delta e_2(k) + k_{2d}^{(r)} \Delta e_2(k) + k_{12}^{(r)} e_1(k) + k_{2o}^{(r)} \end{aligned}$$

where e , Δe , $\Delta^2 e$, δe are the input variables for each controller. This way a similarity between the equations of the conventional digital PID controller (5), (6) and the Sugeno's output functions $f_{1,2}$ in (8) and (9) could be found, where $k_{1,2}$ and $k_{2,1}$ are decoupling coefficients similar to figure 1. In this case, each FPID controller can be considered as a collection of many local PID controllers, which are represented by the Sugeno's functions into the different fuzzy rules. Thus, it is possible to approximate the nonlinear characteristic of the controlled plant.

The novelty here is the fuzzy neural implementation of the PID controllers. There are three inputs in the Sugeno's fuzzy rules and in the fuzzy-neural architecture (figure 2) respectively, equivalent to the three parts of the controller – proportional, integral and derivative.

The Structure of the Proposed Fuzzy-Neural PID Controllers

The proposed FPID controllers with self-tuning parameters are represented by adaptive properties of two

Figure 2. Fuzzy-neural network architecture for each FPID controller

artificial neural networks. The fuzzy-neural structure has 5 layers (figure 2) with particular for fuzzy logic control functionalities.

The first layer simply propagates the input variables towards next layers. Gaussian membership functions with uncertain standard deviation are implemented into the second layer in order to perform the fuzzification of the inputs. The Gaussian membership functions with uncertain deviation are expressed by:

$$(10) \quad \mu_{ij}(x_i) = \exp\left(-\frac{1}{2} \frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right)$$

where c_{ij} is the mean value of the j^{th} fuzzy set for the i^{th} input signal and σ_{ij} is the spread of membership function respectively. The input signals are x_i ($x_i = e, \Delta e, \Delta^2 e$ or δe).

The parameters of the Gaussian function are adjustable. Their values (joined in the parameter a_{ij}) are going to be tuned by the learning algorithm introduced below.

The nodes in the third layer are used to perform precondition matching of the fuzzy logic rules. Hence, the rule nodes R_r should carry out a collection of the membership degrees of the fired fuzzy sets. The fourth layer implements the fuzzy 'product' operation according to (11) or (12) and to integrate the fired rules which have the same consequent.

$$(11) \quad \mu_{1u}^{(r)} = \mu_{e_1}^{(r)} * \mu_{\Delta e_1}^{(r)} * \mu_{\delta e_1}^{(r)} * \mu_{\Delta e_2}^{(r)}$$

$$(12) \quad \mu_{2u}^{(r)} = \mu_{e_2}^{(r)} * \mu_{\Delta e_2}^{(r)} * \mu_{\delta e_2}^{(r)} * \mu_{\Delta e_1}^{(r)}$$

where $\mu_{ij} = \bar{\mu}_{E_r}, \bar{\mu}_{\Delta E_r}, \bar{\mu}_{\delta E_r}$ or $\bar{\mu}_{\Delta^2 E_r}$ specify the membership degrees upon the fired fuzzy sets of the input signals into the $(r)^{th}$ fuzzy rule. They are determined according to (10).

The single node in the fifth layer computes the decision signal out of the network and in this way acts as Sugeno's output defuzzification and it is expressed as a weighted mean value:

$$(13) \quad u_F(k) = \frac{\sum_{r=1}^N f_u^{(r)} \mu_u^{(r)}}{\sum_{r=1}^N \mu_u^{(r)}}$$

where $f_u^{(r)}$ are the Sugeno output functions (8,9) and $\mu_u^{(r)}$ are the corresponded membership degrees of the quantization levels (11,12). This layer also consists of tunable parameters – the coefficients of the PID controllers joined in the parameter β_{jr} .

The Proposed Learning Algorithm

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.

The control action is obtained as a sum of the output signal from the FPID controller u_F and the output signal from the conventional PD controller u_{PD} , working in parallel:

$$(14) \quad u = u_{PD} + u_F.$$

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

$$(15) E = \varepsilon^2 / 2$$

where ε is calculated as the difference $\varepsilon = u - u_F = u_{PD}$, in which u denotes the desired control action and the output u_F is calculated by the fuzzy-neural network. The algorithm performs two-step gradient learning procedure [6]. This algorithm is a kind of a gradient descent method and therefore, it cannot guarantee obtaining the global minimum. Assuming that b_{jr} is an j^{th} adjustable parameter (e.g. the constant $k_p^{(r)}$, $k_i^{(r)}$, $k_d^{(r)}$, or $k_0^{(r)}$) in the Sugeno output function $f_{ur}^{(r)}$ (8) or (9) into the $(r)^{th}$ activated rule, which is represented as a connection for the output neuron in the layer 5, the general parameter learning rule used is [6]

$$(16) \beta_{jr}(k+1) = \beta_{jr}(k) + \eta \left(-\frac{\partial E}{\partial \beta_{jr}} \right) \quad j=1,2,3,4; (r)=1,2,\dots,N$$

where the parameter η is so called learning rate, and the derivative of the error is calculated by partial derivatives:

$$(17) \frac{\partial E}{\partial \beta_{jr}} = \frac{\partial E}{\partial u_F} \cdot \frac{\partial u_F}{\partial f_{ur}^{(r)}} \cdot \frac{\partial f_{ur}^{(r)}}{\partial \beta_{jr}}$$

where:

$$(18) \frac{\partial E}{\partial u_F} = -(u - u_F), \quad \frac{\partial u_F}{\partial f_{ur}^{(r)}} = \frac{\mu_u^{(r)}}{\sum_{r=1}^N \mu_u^{(r)}}, \quad \frac{\partial f_{ur}^{(r)}}{\partial \beta_{jr}} = x_i.$$

After calculating the partial derivatives, the final recurrent equation for each adjustable parameter β_{jr} in the layer 5 for two controllers are:

First FPID controller:

$$(19) \begin{aligned} k_{1p}^{(r)}(k+1) &= k_{1p}^{(r)}(k) + \eta u_{PD_1}(k) \mu_{1u}^{(r)}(k) e_1(k) \\ k_{1i}^{(r)}(k+1) &= k_{1i}^{(r)}(k) + \eta u_{PD_1}(k) \mu_{1u}^{(r)}(k) \delta e_1(k) \\ k_{1d}^{(r)}(k+1) &= k_{1d}^{(r)}(k) + \eta u_{PD_1}(k) \mu_{1u}^{(r)}(k) \Delta e_1(k) \\ k_{21}^{(r)}(k+1) &= k_{21}^{(r)}(k) + \eta u_{PD_1}(k) \mu_{1u}^{(r)}(k) e_2(k) \\ k_{1o}^{(r)}(k+1) &= k_{1o}^{(r)}(k) + \eta u_{PD_1}(k) \mu_{1u}^{(r)}(k) \end{aligned}$$

Second FPID controller:

$$(20) \begin{aligned} k_{2p}^{(r)}(k+1) &= k_{2p}^{(r)}(k) + \eta u_{PD_2}(k) \mu_{2u}^{(r)}(k) e_2(k) \\ k_{2i}^{(r)}(k+1) &= k_{2i}^{(r)}(k) + \eta u_{PD_2}(k) \mu_{2u}^{(r)}(k) \delta e_2(k) \\ k_{2d}^{(r)}(k+1) &= k_{2d}^{(r)}(k) + \eta u_{PD_2}(k) \mu_{2u}^{(r)}(k) \Delta e_2(k) \\ k_{12}^{(r)}(k+1) &= k_{12}^{(r)}(k) + \eta u_{PD_2}(k) \mu_{2u}^{(r)}(k) e_1(k) \\ k_{2o}^{(r)}(k+1) &= k_{2o}^{(r)}(k) + \eta u_{PD_2}(k) \mu_{2u}^{(r)}(k) \end{aligned}$$

The internal two layers 4 and 3 do not contain adjustable parameters or they have constant link weights equal to 1. Therefore the output error E can be propagated back directly to the layer 2, where there are the adjustable parameters a_{jr} . The output error function E is propagated through the links composed by corresponding membership degrees $m_u - m_{ij}$ from the layer 5 to the layer 2. Hence, the learning rule for the second group of the adjustable parameters in the layer 2 can be taken from [6]:

$$(21) \alpha_{jr}(k+1) = \alpha_{jr}(k) + \eta \left(-\frac{\partial E}{\partial \alpha_{jr}} \right)$$

where the derivative of the output error function E is dis-

tributed as follow:

$$(22) \frac{\partial E}{\partial \alpha_{jr}} = \frac{\partial E}{\partial u_F} \cdot \frac{\partial u_F}{\partial \mu_{ij}} \cdot \frac{\partial \mu_{ij}}{\partial \alpha_{jr}}.$$

The first partial derivative has been calculated in (18) and the second one is:

$$(23) \frac{\partial u_F}{\partial \mu_{ij}} = \frac{f_u^{(r)} - u_F}{\sum_{r=1}^N \mu_u^{(r)}}.$$

There are two possibilities to calculate the third partial derivative in (22) for each adjustable parameter according to the equations (10) for each Gaussian membership function ((24) for the center c_{ij} or (25) for the spreads σ_{ij} of Gaussian membership function):

$$(24) \frac{\partial \mu_{ij}}{\partial c_{ij}} = \frac{[x_i(k) - c_{ij}]}{\sigma_{ij}^2}$$

$$(25) \frac{\partial \mu_{ij}}{\partial \sigma_{ij}} = \frac{[x_i(k) - c_{ij}]^2}{\sigma_{ij}^3}.$$

After calculating the partial derivatives, the final recurrent equation for each adjustable parameter for i^{th} input variable and its j^{th} fuzzy set is:

$$(26) c_{ij}(k+1) = c_{ij}(k) + \eta u_{PD} \mu_u^{(r)}(k) [f_u^{(r)}(k) - u_F(k)] \frac{[x_i(k) - c_{ij}(k)]}{\sigma_{ij}^2(k)}$$

$$(27) \sigma_{ij}(k+1) = \sigma_{ij}(k) + \eta u_{PD} \mu_u^{(r)}(k) [f_u^{(r)}(k) - u_F(k)] \frac{[x_i(k) - c_{ij}(k)]^2}{\sigma_{ij}^3(k)}$$

The final recurrent equations (19), (20), (26) and (27) are used in the software implementation of the developed algorithm for the FPID controllers.

Simulation Results

The investigations have been carried out by simulations of the decoupling fuzzy algorithm in MATLAB&Simulink environment with a nonlinear plant – Quadruple Tank System, described in [2]. The control algorithm is programmed in MATLAB code and is implemented as a MATLAB function block into the Simulink model for simulations.

The quadruple tank plant (QTP) is a multivariable laboratory plant of interconnected tanks with nonlinear dynamics and subject to state and input constraints. A scheme of this plant is presented in figure 3. The QTP consists of four interconnected water tanks, a basin of water, and two pumps. The output of each pump is split between a lower tank and the opposite upper tank. Water flows through orifices at the bottoms of each upper tank into the corresponding lower tanks. Water also flows through orifices at the bottoms of the lower tanks into the basin. The goal of this project is to design a multivariable digital tracking system that can follow step commands for the water heights in the lower tanks [2].

The experimental model has two inputs (pump speeds)

Figure 3. Basic view of the quadruple-tank process

Figure 4. Level control results with classical PID controllers

which can be manipulated to control the two outputs (tank levels). The system exhibits multivariable dynamics because each of the pumps affects both of the outputs. The system has an adjustable multivariable zero that can be set to a right-half or left-half plane value by changing the valve settings of the experiment. Unmeasured disturbances can be applied by pumping water out of the top tanks and into the lower reservoir [2]. The described laboratory quadruple-tank process consists of four interconnected water tanks and two pumps. The first principle mathematical model for this process using mass balances and Bernoulli's law is [2].

$$(28) \quad \frac{dh_1}{dt} = -\frac{a_1}{A_1} \sqrt{2gh_1} + \frac{a_3}{A_1} \sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1} v_1$$

$$(29) \quad \frac{dh_2}{dt} = -\frac{a_2}{A_2} \sqrt{2gh_2} + \frac{a_4}{A_2} \sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2} v_2$$

$$(30) \quad \frac{dh_3}{dt} = -\frac{a_3}{A_3} \sqrt{2gh_3} + \frac{(1-\gamma_1)k_1}{A_3} v_1$$

$$(31) \quad \frac{dh_4}{dt} = -\frac{a_4}{A_4} \sqrt{2gh_4} + \frac{(1-\gamma_2)k_2}{A_4} v_2$$

where γ_i is the flow distribution to lower and diagonal upper tank, A_i is the cross-section area, a_i is the outlet hole cross-section and h_i is the water level in tank i respectively. There are two inputs and two outputs (controlled variables) in quadruple tank system, and control objective is to maintain water level in lower tanks at their setpoints with the manipulation of water flow with two pumps. The process inputs are v_1 and v_2 and the outputs are $y_1=k_1h_1$ and $y_2=k_2h_2$

[2]. The voltage applied to the *Pump i* is v_i and the corresponding flow is $Q_i=k_i v_i$. The parameters $\gamma_1, \gamma_2 \in (0,1)$ are valves setting for the distribution of flow to lower and upper diagonal tank respectively. The flow to *Tank 1* is $\gamma_1 k_1 v_1$ and the flow to *Tank 4* is $(1-\gamma_1)k_1 v_1$ and similarly for *Tank 2* and *Tank 3*. Here g denotes the gravity acceleration.

The simulation results of the quadruple-tank process with the proposed decoupling fuzzy-neural PID and a classical PID control are presented graphically. The results illustrate the performance of both control strategies. Parameters of the experimental studies are given in *table 1*.

Figure 4 shows level control results with classical PID controllers and *figure 5* shows level control results with the proposed decoupling FPID controllers. The different quantities are marked as follows: the two set points – s_1 and s_2 , system outputs – y_1 and y_2 , and control actions q_a and q_b .

The advantage of the discussed tuned fuzzy controller with decoupling effect is that it improves the system performance. Simulation results confirm the effectiveness of the proposed control system.

Conclusions

This paper is focused on design of an algorithm for decoupling fuzzy-neural PID control and its application to control of a quadruple tank system a TITO plant. To handle

Figure 5. Level control results with decoupling FPID controllers

Table 1. Experimental parameters

Parameters	Value
$A_1, A_2, A_3, A_4 \text{ m}^2$	0.06
$a_1, a_2 \text{ cm}^2$	1.3104; 1.5074
$a_3, a_4 \text{ cm}^2$	9.2673; 8.8164
$g \text{ m/s}^2$	9.81
$h_{10}, h_{20} \text{ m}$	0.5; 0.6
$h_{30}, h_{40} \text{ m}$	0.7; 0.8
γ_1, γ_2	0.3; 0.4
k_1, k_2	1

the nonlinearities, a Takagi–Sugeno fuzzy – neural network is suggested as a means to control the plant with nonlinearities depending on the operating region. The coupling effect is added into the main fuzzy controller for each step to improve the system performance. The proposed technique has been tested and evaluated using this simulated industrial plant.

The advantage of fuzzy neural network control is that the controller can be taken as a construction through many local linear parallel distributed controllers, which usually have analytical solutions. Hence, the control actions of this nonlinear fuzzy–neural controller can be obtained analytically avoiding the time consuming numerical search procedures, using similarity with conventional PID controller.

Furthermore, the obtained control actions based on the local incremental controllers contain integral action, which can naturally eliminate steady state system errors.

The advantages of the discussed tuned fuzzy controller are that it improves performance, decreases complexity, reduces computational time and needs less memory.

References

1. Astrom, K. and B. Wittenmark. Computer Controlled Systems. 3 Ed., Prentice Hall, 1997.
2. Digital Control of a Quadruple-Tank Process. ELE/MCE 503 Final Project, Fall 2012. <http://egr.uri.edu>.
3. Glatfelder, A. H., W. Schaufelberger. Control Systems with Input and Output Constraints. Springer, 2003.
4. Gomi, H. and M. Kawato. Neural Network Control for a Closed-Loop System Using Feedback-Error-Learning. – *Neural Networks*, 6, 1993.
5. Takagi, H. and M. Sugeno. Fuzzy Identification of Systems and its Applications to Modeling and Control. – *IEEE Trans. on Systems, Man., and Cybern.*, SMC-15(1), 1985, 116-132.
6. Petrov, M., I. Ganchev, A. Taneva. Fuzzy PID Control of Nonlinear Plants. Proceedings of the IEEE International Symposium on “Intelligent Systems”, Varna, Bulgaria, 10-13 September 2002, IEEE Catalogue Number 02EX499, ISBN 0-7803-7601-3, 1, 30-35.
7. Ahmed, S., I. Ganchev, A. Taneva, M. Petrov. Decoupling Neuro-Fuzzy Model Predictive Controllers Applied to Quadruple Tanks. 2016 IEEE 8th International Conference on Intelligent Systems, 610-615.
8. Chia-Feng, Juang and Chin-Teng Lin. An On-Line Self-Constructing Neural Fuzzy Inference Network and Its Applications. – *IEEE trans. Fuzzy System*, 6, 1999, No. 2, 13-31.
9. Yang, J. R. ANFIS: Adaptive-Network-Based Fuzzy Inference System. – *IEEE Transactions on Systems, Man, and Cybernetic*, 23, May/June 1993, No. 3, 665-685.
10. Nurnberger, A., D. Nauck and R. Kruse. Neuro-fuzzy Control Based on the Nefcon-model: Recent Developments. – *Soft Computing*, 2, 1999, 168-182.
11. Azeem, Mohammad Fazle, Madasu Hanmandlu, Nesar Ahmed. Structure Identification of Generalized Adaptive Neuro-Fuzzy Inference Systems. – *IEEE Trans. Fuzzy System.*, 11, Oct. 2003, No. 5, 666-681.
12. Shie-Jue, Lee and Chen-Sen Ouyang. A Neuro-fuzzy System Modeling with Self-Constructing Rule Generation and Hybrid SVD-Based Learning. – *IEEE Trans. Fuzzy System.*, 11, June 2003, No. 3, 341-353.
13. Chin-Teng Lin and C. S. George Lee. Reinforcement Structure/Parameter Learning for Neural Network Based Fuzzy Logic Control Systems. – *IEEE Trans. Fuzzy System.*, 2, February 1994, No. 1, 46-73.
14. Castellano, G. and A. M. Fanelli. Fuzzy Inference and Rule Extraction and Using Neural Network. – *Neural Network World Journal*, 3, 2000, 361-371.
15. Chin-Teng Lin and Ya-Ching Lu. A Neural Fuzzy System with Fuzzy Supervised Learning. – *IEEE Transactions on Systems, Man, and Cybernetic-Part B*, 26, Oct. 1996, No. 5, 744-763.

Manuscript received on 11.11.2016

Ivan Ganchev is an associate professor in Process Control at the Technical University – Sofia, Plovdiv branch. He received his MEng degree in Industrial Automation from the University of Food Technologies in 1983 and Ph.D. from the Technical University – Sofia in 2002. His research interests are Process Control, Adaptive Control, and Automatic Tuning of controllers.

Contact:
Technical University of Sofia
Plovdiv branch
Control Systems Department
e-mail: ganchev@tu-plovdiv.bg

Sevil Ahmed received her PhD in Artificial Intelligence from Technical University of Sofia, Bulgaria in 2013. She is currently Assistant Professor at the Department of Control Systems, Technical University of Sofia, Branch Plovdiv, Bulgaria. Her current research interests include adaptive control, fuzzy-neural structures, robotics, sensor networks, and model predictive control.

Contact:
Technical University of Sofia
Plovdiv branch
Control Systems Department
e-mail: sevil.ahmed@tu-plovdiv.bg

Albena Taneva is an associate professor in Process Control at the Technical University – Sofia, Plovdiv branch. She received her MEng degree in Automatics and systems from the Technical University Sofia, Plovdiv branch in 1998 and Ph.D. from the Technical University – Sofia in 2005. Her research interests are

Computer Integrated Manufacturing, Process Control, Fuzzy Logic Controllers, Programmable Logic Controllers, and Distributed Control Networks.

*Contact:
Technical University of Sofia
Plovdiv branch
Control Systems Department
e-mail: altaneva@tu-plovdiv.bg*

Michail Petrov is a professor in Industrial Automation at the Technical University – Sofia, Plovdiv branch. He received MSc degree in Technical Cybernetics from Czech Technical University in Prague, Czech Republic and a Ph.D. degree in Industrial Automation from Mining University Sofia. His professional and scientific research interests include control systems design, fuzzy control systems and model predictive control.

*Contacts:
Technical University of Sofia,
Plovdiv branch
Control Systems Department,
e-mail: mpetrov@tu-plovdiv.bg*