Appl. Math. and Comp. Sci., 1997, vol.7, No.1, 171–184

FAST-LEARNING NEURO-FUZZY PID-CONTROLLER WITH MINIMAL NUMBER OF FUZZY REGIONS

Andrzej PIEGAT*, Marcin PLUCIŃSKI*

Some problems connected with learning neuro-fuzzy controllers are presented and then the concept of a neuro-fuzzy PID-controller with a minimal number of fuzzy regions is discussed. The controller has soft performance, which distinguishes it from classical fuzzy controllers. It is characterized by short training time and genetic search algorithms are not required. The advantages mentioned above were achieved owing to the application of linear AND and OR operators. Learning results of the controller in a system with reference model are presented in the final part of the paper.

1. Introduction

A great advantage of fuzzy logic consists in a possibility of transforming the human knowledge about the real system into mathematical form of a fuzzy model describing inputs/output causalities of the system.

The first fuzzy models of technical systems were created based only on the system expert's knowledge (Abel, 1991). Then it became evident that the human knowledge suffices only for modelling simplest systems with at most two inputs. Learning the inputs/output causalities and real-time control of three-input plants by a man is practically impossible, because of his limited perception and reaction speed.

It was also claimed that the knowledge of various experts about the same system can be different, because it is burdened with subjectivity and observation errors of a particular expert. Achieving the objective system knowledge would be possible by learning with inputs/output measurement data. Since neural networks (NNs) have this ability, a fusion of fuzzy logic with them was made, which resulted in creation of neuro-fuzzy networks (NFNs), (Grant and De Bruijn, 1993; Horikawa *et al.*, 1992; Masuoka *et al.*, 1990; Preuss and Tresp, 1994; Takagi, 1990). Great expectations were connected with NFNs.

NFNs were expected to combine *the advantages of fuzzy logic* such as

- operating with qualitative structured knowledge in the form of inference rules IF A THEN B,
- a possibility of understanding and interpreting the fuzzy model by a man,

* Department of Computer Science and Engineering, Technical University of Szczecin, 71-065 Szczecin, Al. Piastów 41, Poland, e-mail: mapluc@dedal.man.szczecin.pl.

- a possibility of introducing the initial qualitative knowledge about the system into a fuzzy model,
- a possibility of supplementing a fuzzy model with the use of knowledge acquired during the system operation,

with *the advantages of neural networks* such as:

- the ability to learn on the basis of numerical inputs/output measurement data,
- objectivity of the learning process (independence from subjective errors of a system expert).

However, investigations of NFNs brought about some disappointments (Preuss and Ockel, 1995). It turns out that NFNs adopt not only the advantages of their components but also their disadvantages, e.g. difficulties in the learning process which are related to getting stuck in local minima. The learning process is usually very long and it should be repeated for several starting points (genetic algorithms) and nevertheless it often fails.

The NFNs were expected to allow for extraction of inference rules IF A THEN B, i.e. qualitative knowledge about the system. However, it turns out, in practice, that the learning results do not have integer (0 or 1) coefficients for particular rules but only fractional ones (Pfeifer, 1995). This makes it impossible to evaluate which rule is univocally true and forces one to round coefficients, which decreases the model accuracy. Also, the application of RBF-networks for fuzzy systems modelling did not produce expected results (Preuss and Ockel, 1995).

Fuzzy and neuro-fuzzy controllers are often characterized by a very violent action which increases control costs and complicates set mechanisms of plants (Piegat and Baccari, 1995). The above-mentioned disadvantages of NFNs made some scientists go over the creation of fuzzy models with direct knowledge extraction from the system measurement data (Babuška and Verbruggen, 1995; Baldwin and Martin, 1995; Preuss and Ockel, 1995). What is the reason of all the said problems with NFNs?

2. Linear AND and OR Operators

Investigations carried out by the authors have shown that the main reason behind the difficulties in learning NFNs is using classical nonlinear fuzzy-set intersection (AND) and union (OR) operators (Piegat, 1996). For the intersection of two fuzzy sets A and B the following operators of t-norm are used most often (Wang, 1994):

- MINIMUM:

$$m_{A\cap B}(x) = \min\left[m_A(x),\, m_B(x)\right] \tag{1}$$

- ALGEBRAIC PRODUCT:

$$m_{A\cap B}(x) = m_A(x)m_B(x)$$

where $m_A(x)$ and $m_B(x)$ are the membership functions of the fuzzy sets A and B, respectively. The following union operators of two fuzzy sets are widely used.

- MAXIMUM:

$$m_{A\cup B}(x) = \max\Big[m_A(x),\, m_B(x)\Big] \tag{2}$$

- BOUNDED SUM:

$$m_{A\cup B}(x) = \min\Big[1,\, m_A(x) + m_B(x)\Big]$$

Let us now investigate the system with two inputs and one output shown in Fig. 1.

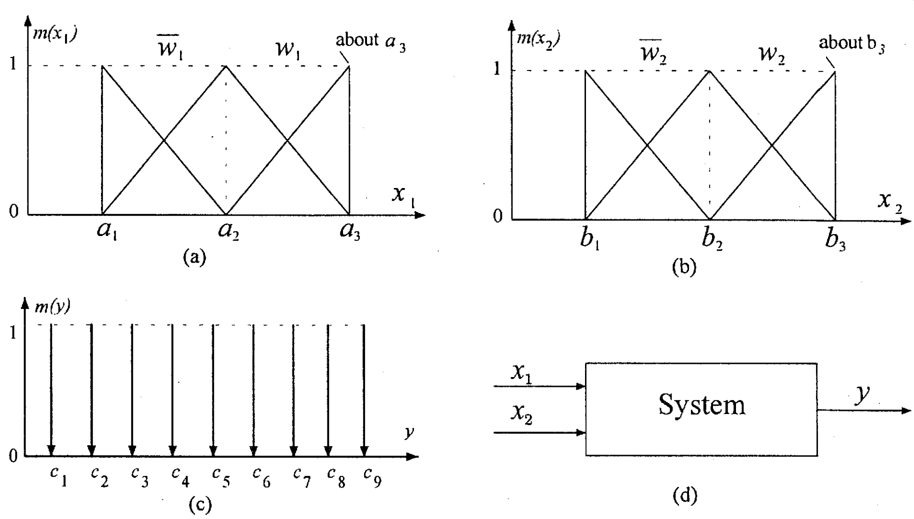

Fig. 1. Two-input-one-output system (d), membership functions of the inputs x_1 (a) and x_2 (b), and the output y (c).

Let the input-output causalities be described by the following inference rules:

$$\begin{aligned}
&R1: \text{ IF } (x_1 \text{ is about } a_1) \text{ AND } (x_2 \text{ is about } b_1) \text{ THEN } (y \text{ is about } c_1)\\
&R2: \text{ IF } (x_1 \text{ is about } a_1) \text{ AND } (x_2 \text{ is about } b_2) \text{ THEN } (y \text{ is about } c_2)\\
&R3: \text{ IF } (x_1 \text{ is about } a_1) \text{ AND } (x_2 \text{ is about } b_3) \text{ THEN } (y \text{ is about } c_3)\\
&R4: \text{ IF } (x_1 \text{ is about } a_2) \text{ AND } (x_2 \text{ is about } b_1) \text{ THEN } (y \text{ is about } c_4)\\
&R5: \text{ IF } (x_1 \text{ is about } a_2) \text{ AND } (x_2 \text{ is about } b_2) \text{ THEN } (y \text{ is about } c_5)\\
&R6: \text{ IF } (x_1 \text{ is about } a_2) \text{ AND } (x_2 \text{ is about } b_3) \text{ THEN } (y \text{ is about } c_6)\\
&R7: \text{ IF } (x_1 \text{ is about } a_3) \text{ AND } (x_2 \text{ is about } b_1) \text{ THEN } (y \text{ is about } c_7)\\
&R8: \text{ IF } (x_1 \text{ is about } a_3) \text{ AND } (x_2 \text{ is about } b_2) \text{ THEN } (y \text{ is about } c_8)\\
&R9: \text{ IF } (x_1 \text{ is about } a_3) \text{ AND } (x_2 \text{ is about } b_3) \text{ THEN } (y \text{ is about } c_9)
\end{aligned} \tag{3}$$

The membership functions from Fig. 1 and the rules (3) make a full fuzzy model of the system. It should be noted that the model has 15 degrees of freedom which/must be appropriately tuned to get a high accuracy. These are the coefficients a_1, a_2, a_3, b_1, b_2, b_3 determining the input membership functions (Fig. 1) and $c_1, \dots, c_9$ for the output singleton positions.

Since the performance of a fuzzy system depends on the region in which the actual inputs and output of the system remain, we introduce the following logical functions:

$$w_1(x_1) = \begin{cases} 1 & \text{if } a_2 < x_1 \le a_3 \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad w_2(x_2) = \begin{cases} 1 & \text{if } b_2 < x_2 \le b_3 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The membership functions of the linguistic values used in the rules (3) can be expressed in the form

$$\begin{cases} m_{a1}(x_1) = \left(\dfrac{a_2 - x_1}{a_2 - a_1}\right)\overline{w}_1, \quad m_{a2}(x_1) = 1 - m_{a1}(x_1) - m_{a3}(x_1) \\ m_{a3}(x_1) = \left(\dfrac{x_1 - a_2}{a_3 - a_2}\right)w_1 \\ m_{b1}(x_2) = \left(\dfrac{b_2 - x_2}{b_2 - b_1}\right)\overline{w}_2, \quad m_{b2}(x_2) = 1 - m_{b1}(x_2) - m_{b3}(x_2) \\ m_{b3}(x_2) = \left(\dfrac{x_2 - b_2}{b_3 - b_2}\right)w_2 \end{cases} \tag{5}$$

It can be proved by the output calculation for the network in Fig. 2 (Piegat, 1996) that with ALGEBRAIC PRODUCT as the union operator in rules (3) the fuzzy model performs the mapping of the inputs x_1 and x_2 into the output y of the form

$$\begin{aligned} y = E_0 &+ (E_1 + E_2x_1)\overline{w}_1 + (E_3 + E_4x_1)w_1 + (E_5 + E_6x_2)\overline{w}_2 \\ &+ (E_7 + E_8x_2)w_2 + (E_9 + E_{10}x_1 + E_{11}x_2 + E_{12}x_1x_2)\overline{w}_1\overline{w}_2 \\ &+ (E_{13} + E_{14}x_1 + E_{15}x_2 + E_{16}x_1x_2)\overline{w}_1 w_2 \\ &+ (E_{17} + E_{18}x_1 + E_{19}x_2 + E_{20}x_1x_2)w_1\overline{w}_2 \\ &+ (E_{21} + E_{22}x_1 + E_{23}x_2 + E_{24}x_1x_2)w_1w_2 \end{aligned} \tag{6}$$

In the mapping (6), E_i are constant coefficients. Their values depend on parameters a_i, b_i, c_i of membership functions. From formula (6) it follows that the surface of the mapping $(x_1, x_2) \rightarrow y$ is nonlinear and its form and position depend on up to 25 coefficients E_i which must be determined in the learning process of the model with input/output data of the real system. However, in the fuzzy model (rules (3) and membership functions in Fig. 1) we have only 15 degrees of freedom at our disposal (the parameters a_i, b_i, c_i). In this situation (a shortage of degrees of freedom), can the fuzzy model surface be adapted to the real system surface? The answer is: No or with great difficulties.

The adaptation can usually be performed only after multiple starts from various initial points (genetic algorithms) and the accuracy of the model achieved is not high. If instead of ALGEBRAIC PRODUCT other operators of t-norm (Wang, 1994), e.g. the MINIMUM operator, are used, the complexity of the mapping $(x_1, x_2) \rightarrow y$ and the number of coefficients increase. This explains the difficulties in learning NFNs reported in the literature and experienced also by the authors.

The question arises: How could the learning effectivity of NFNs be increased? Theoretical and experimental investigation have shown (Piegat, 1995; 1996) that the adaptation (learning) effectivity increases substantially when linear intersection and union operators are used which are defined as follows:

- The linear intersection operator of fuzzy sets $A_1 \cap \ldots \cap A_n$, i.e. ARITHMETIC MEAN (MEAN):

$$\begin{aligned} &\forall x : m_{A_1\cap\ldots\cap A_n}(x) = \frac{1}{n}\sum_{i=1}^{n} m_{A_i}(x) \\ &\mathrm{MEAN} : [0,\infty)^n \rightarrow [0,\infty), \quad x \in X \end{aligned} \tag{7}$$

- The linear union operator of fuzzy sets $A_1 \cup \ldots \cup A_n$, i.e. ARITHMETIC SUM (SUM):

$$\begin{aligned} &\forall x : m_{A_1\cup\ldots\cup A_n}(x) = \sum_{i=1}^{n} m_{A_i}(x) \\ &\mathrm{SUM} : [0,\infty)^n \rightarrow [0,\infty), \quad x \in X \end{aligned} \tag{8}$$

where X is the universe of discourse.

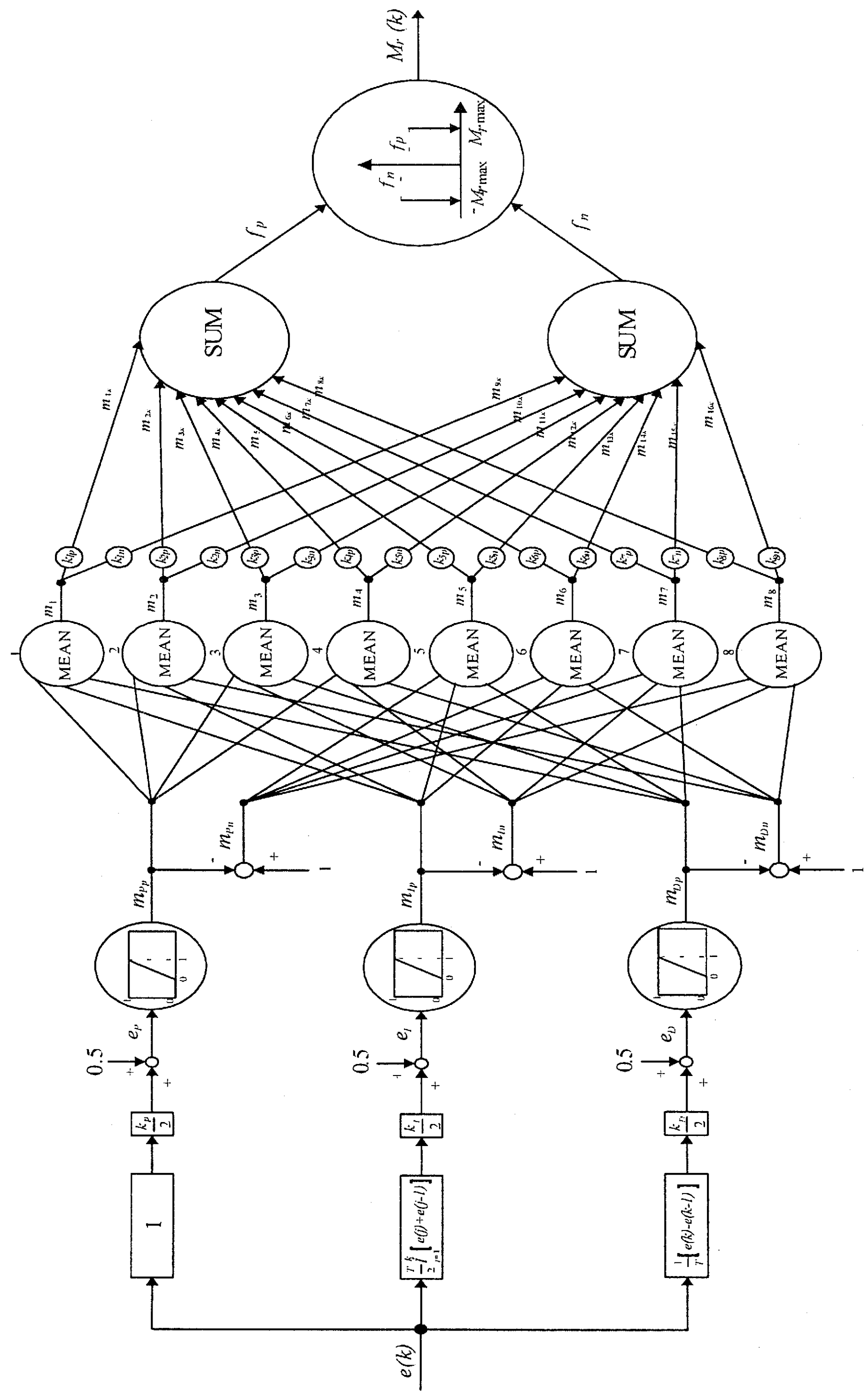

Fig. 2. Structure of the neuro-fuzzy PID controller with 8 fuzzy regions.

The application of these operators induces a transformation of fuzzy sets being the basis of fuzzy modelling into fuzzy-bag sets (Driankov *et al.*, 1993; Piegat, 1996; Zimmerman, 1996). To explain the notion of the "fuzzy-bag set", the notions "bag" and "fuzzy bag" have to be clarified.

A bag A is any collection of elements drawn from the universe of discourse X. Multiple copies of the same element are allowed. A fuzzy bag B is the bag of pairs: (element x, membership degree of x to the bag B). Multiple copies of the same element x_i in the pairs are allowed and membership degrees of the copies x_i in different pairs may be different. Such a situation occurs in defuzzification in learning neuro-fuzzy networks when two or more simple rules activate the same singleton with different membership degrees. When classical nonlinear operators, e.g. MAX, are used, some information is lost and the neuro-fuzzy model accuracy is worse. A fuzzy bag B can be characterized as follows:

$$B = \left\{ \left(x, \mu_B(x) \right) \Big| \, x \in X \right\} \quad \text{with} \quad \mu_B : X \to [0,1]$$

A fuzzy-bag set $C(B)$ is a set achieved by transformation of the original fuzzy bag B containing multiple elements x_i in its pairs $(x, \mu_B(x))$ into a fuzzy set $C(B)$ without multiple elements. The membership degree of the elements x_k to a fuzzy-bag set $C(B)$ can be defined by the membership function μ_C:

$$\mu_C(x_k) = \sum_{j=1}^{m} \mu_{Bj}(x_k), \quad \mu_C : X \to [0, \infty]$$

where m is the multiplicity of the element x_k in pairs $(x, \mu_B(x))$ of the original fuzzy bag B. To calculate the membership to a fuzzy-bag set, an operator with additivity feature must be used. But the classical operators do not allow for additivity.

The main advantage of the linear operators MEAN and SUM is that they give a linear interpolation surface connecting the points of the input-output space defined by the rules of a fuzzy model, whereas the interpolation achieved with classical operators is nonlinear. As will be shown, it is of great importance. A fuzzy-bag set is a more general notion than a fuzzy set.

With the linear operators AND and OR in the model (3)–(5), a segment-linear surface of the fuzzy model was achieved by successively calculating the elements' outputs in the network of Fig. 2:

$$y = E_0 + (E_1 + E_2 x_1)\overline{w}_1 + (E_3 + E_4 x_1)w_1 + (E_5 + E_6 x_2)\overline{w}_2 + (E_7 + E_8 x_2)w_2 \quad (9)$$

For complete definition of this surface, only nine E_i coefficients should be determined. And the fuzzy model (3) has 15 degrees of freedom at its disposal. So we have here an excess of degrees of freedom, which accelerates considerably learning the NFN and increases the final accuracy. This was confirmed by experiments made by the authors and their collaborators (Piegat *et al.*, 1996). In what follows, a nonlinear fuzzy PID controller working with linear AND/OR operators will be presented.

3. Structure of the Neuro-Fuzzy PID Controller with Minimal Number of Fuzzy Regions

The controller gets the actual value of the control error e and calculates the values of the numerical integral e_I and of the numerical derivative e_D of the error (Fig. 2). The signals e_P, e_I and e_D are the inputs of the fuzzy part of the controller.

The controller realizes the following rule base:

$$\begin{aligned}
&R1: \text{ IF } \Big[(e_p = p) \text{ AND } (e_i = p) \text{ AND } (e_d = p)\Big] \text{ THEN } \Big[M_r = M_{r\,\max}(k_{1p} \text{ OR } - k_{1n})\Big] \\
&R2: \text{ IF } \Big[(e_p = p) \text{ AND } (e_i = p) \text{ AND } (e_d = n)\Big] \text{ THEN } \Big[M_r = M_{r\,\max}(k_{2p} \text{ OR } - k_{2n})\Big] \\
&R3: \text{ IF } \Big[(e_p = p) \text{ AND } (e_i = n) \text{ AND } (e_d = p)\Big] \text{ THEN } \Big[M_r = M_{r\,\max}(k_{3p} \text{ OR } - k_{3n})\Big] \\
&R4: \text{ IF } \Big[(e_p = p) \text{ AND } (e_i = n) \text{ AND } (e_d = n)\Big] \text{ THEN } \Big[M_r = M_{r\,\max}(k_{4p} \text{ OR } - k_{4n})\Big] \\
&R5: \text{ IF } \Big[(e_p = n) \text{ AND } (e_i = p) \text{ AND } (e_d = p)\Big] \text{ THEN } \Big[M_r = M_{r\,\max}(k_{5p} \text{ OR } - k_{5n})\Big] \\
&R6: \text{ IF } \Big[(e_p = n) \text{ AND } (e_i = p) \text{ AND } (e_d = n)\Big] \text{ THEN } \Big[M_r = M_{r\,\max}(k_{6p} \text{ OR } - k_{6n})\Big] \\
&R7: \text{ IF } \Big[(e_p = n) \text{ AND } (e_i = n) \text{ AND } (e_d = p)\Big] \text{ THEN } \Big[M_r = M_{r\,\max}(k_{7p} \text{ OR } - k_{7n})\Big] \\
&R8: \text{ IF } \Big[(e_p = n) \text{ AND } (e_i = n) \text{ AND } (e_d = n)\Big] \text{ THEN } \Big[M_r = M_{r\,\max}(k_{8p} \text{ OR } - k_{8n})\Big]
\end{aligned} \tag{10}$$

The symbols 'p' and 'n' mean 'positive' and 'negative', respectively. The coefficients k_{ip} and k_{in}, tuned in the learning process, determine how much the conclusion of the rule R_i should activate the positive or negative output singleton ($M_{r\,\max}$ or $-M_{r\,\max}$). The values of these coefficients can vary within the interval $[0, 1]$. $M_{r\,\max}$ means the maximal output value of the controller. The proposed controller works in the space with a minimal number of fuzzy regions (Fig. 3).

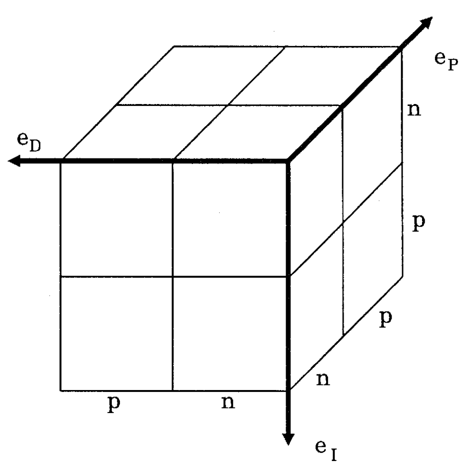

Fig. 3. Fuzzy regions of the neuro-fuzzy PID controller.

Such a number of regions can be achieved only when fuzzification of individual controller inputs e_P, e_I and e_D is realized merely with two linguistic values: positive and negative (Fig. 4).

Defuzzification is carried out with singletons, according to

$$M_r(k) = \left(\frac{f_p - f_n}{f_p + f_n}\right) M_{r\,\max} \tag{11}$$

where f_p and f_n denote the activation degrees of the positive and negative output singletons resulting from the rule conclusions (10), respectively.

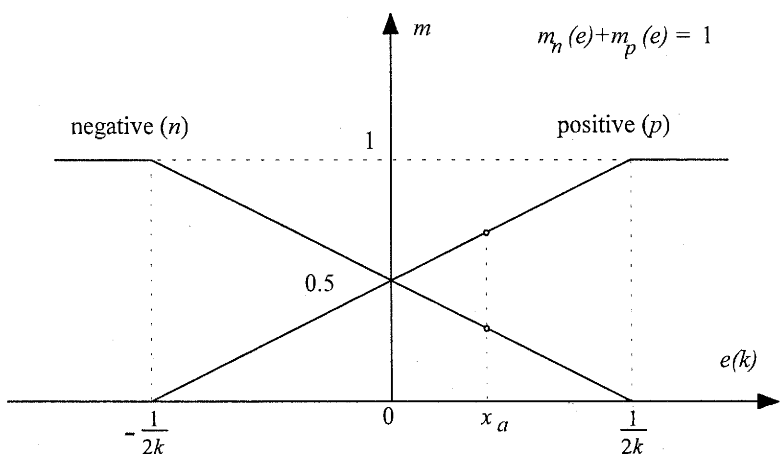

Fig. 4. Inputs fuzzification in the neuro-fuzzy PID controller.

The controller has 11 degrees of freedom (K_P, K_I, K_D, $k_{1p}, \ldots, k_{8p}$) undergoing the learning process. However, because of the following specific symmetry condition of the controller:

$$k_{ip} = 1 - k_{(9-i)p}, \quad k_{in} = 1 - k_{ip}, \quad k_{1p} = 1, \quad k_{8p} = 0 \tag{12}$$

the number of tuned parameters could be reduced to 6 (K_P, K_I, K_D, k_{2p}, k_{3p}, k_{4p}).

4. Experiment

The process of learning the NF-PID controller was realized in the so-called direct structure (Korbicz *et al.*, 1994), see Fig. 5.

Learning of this type, rarely found in the literature, was performed with the assumption of the plant-model knowledge (the motion model of an underwater vehicle). In the learning process with control system reference model no human knowledge (a system expert's knowledge) is needed. The plant model, described more accurately in (Piegat and Pluciński, 1995), is shown in Fig. 6.

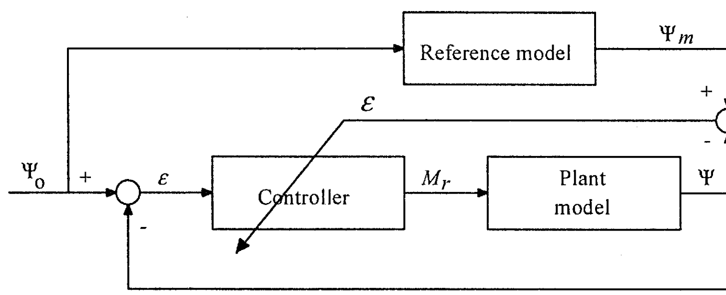

Fig. 5. Structure of the learning system of the neuro-fuzzy PID controller.

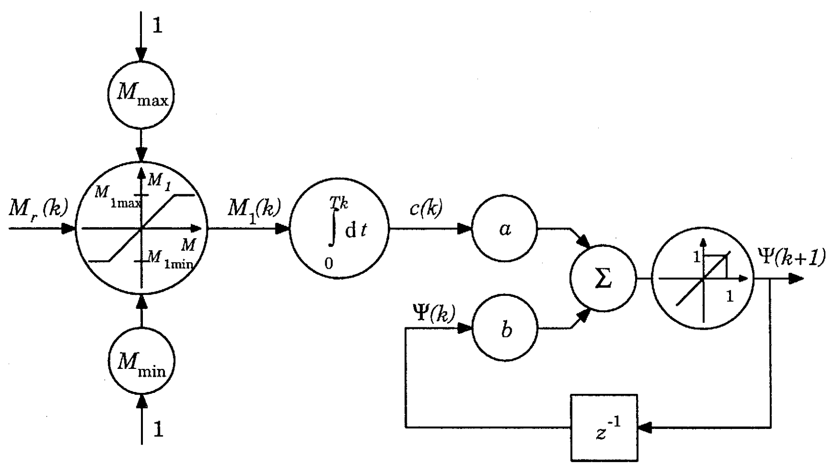

Fig. 6. Neural model of the plant (an underwater vehicle).

The integration neuron appearing in this model, realized numerically, is rare in neural networks and was not found by the authors in the literature. It was proved that the error ε can be also backpropagated through the integration neuron.

As the reference model for the whole control system, the following model was applied:

$$G_r(s) = \frac{\Psi_m(s)}{\Psi_0(s)} = \frac{1}{1+0.2s} \tag{13}$$

Figure 7 shows a triangle reference signal ψ_0 and the resulting reference model output ψ_m. These signals were used in learning the NF-PID controller with back-propagation method with momentum, (Korbicz *et al.*, 1994; Tadeusiewicz, 1993).

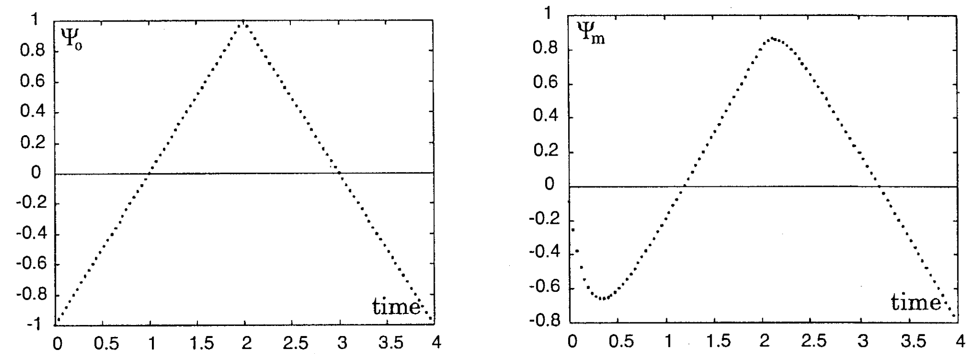

Fig. 7. Signals used in learning the neuro-fuzzy PID controller.

Figure 8 shows the convergence of the learning error $\varepsilon = \psi_m - \psi$. It can be seen that stable convergence of the learning process was not achieved. But the process came to a repeated oscillation cycle with minimum $\varepsilon = 0.077$ (the initial error was $\varepsilon = 0.36$). The minimal error value was achieved after 21696 learning cycles without application of genetic algorithms search. The learning time was about two hours (PC 486 DX2, 66 MHz).

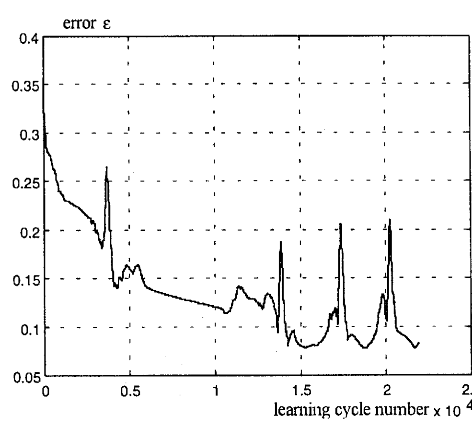

Fig. 8. Variations of the learning error for the NF-PID controller.

Figure 9 shows the response of the reference model ψ_m and of the control system to the unit step as the reference signal ψ_0 before learning (the initial parameters of the controller: $k_P = 1$, $k_I = 0.5$, $k_D = 2$, $k_{2p} = k_{3p} = k_{4p} = 1$), and after learning (the determined parameters: $k_P = 0.596612$, $k_I = 1.76949$, $k_D = 2{,}24621$, $k_{2p} = 0.52887$, $k_{3p} = 0.655561, k_{4p} = 0.679226$, other coefficients according to (12)). Such results were achieved in the second learning experiment. The authors also investigated other set intersection and union operators (nonlinear: MINIMUM and MAXIMUM). In this case, the learning process was longer and, in spite of using various starting points (the initial parameters of the controller), the foregoing good results were not achieved.

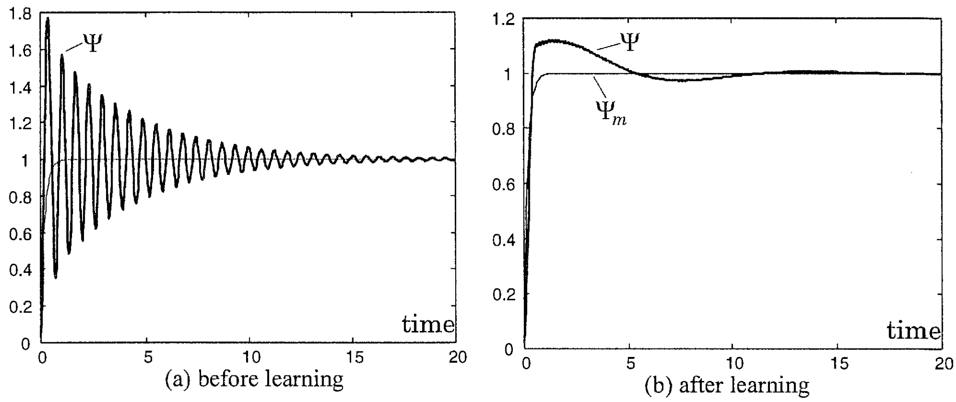

Fig. 9 Step response ψ of the control system with the NF-PID controller and of the reference model ψ_m.

Figure 10 shows the responses of the reference model ψ_m and of the control system with learned controller to sinusoidal and triangle signals with double frequency (in relation to the learning input signal).

As can be concluded from Figs. 9 and 10, the control system with the trained NF-PID controller imitates the reference model with good accuracy also in tests with signals different from those used in learning.

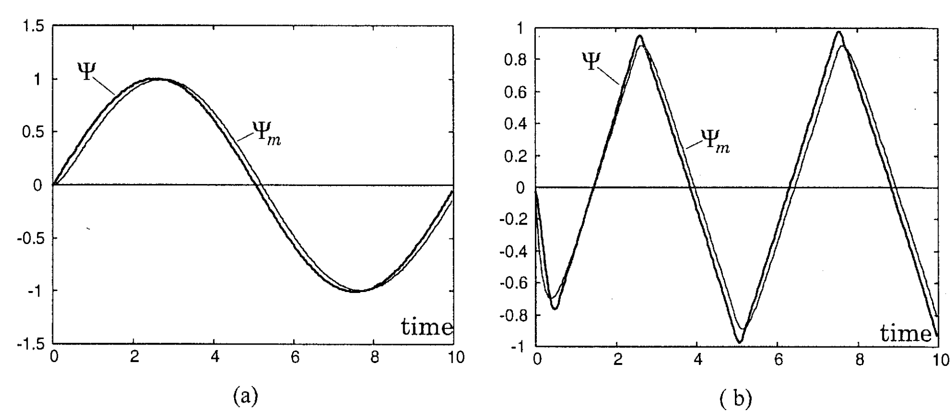

Fig. 10 Comparison of responses of the reference model ψ_m and of the control system with the learned NF-PID controller on the sinusoidal (a) and triangle (b) signals.

In this section, only the results of one experiment are shown. The authors carried out many other experiments which confirmed the results described here. Therefore, the authors now in their investigations use only linear operators in learning neuro-fuzzy networks.

5. Conclusions

The structure of the neuro-fuzzy PID-controller presented in the paper is effective and capable of fast and accurate learning. This effectiveness was achieved owing to the application of linear fuzzy-set intersection and union operators (MEAN and SUM). Using these operators considerably reduces the complexity level of the control surface realized by a neuro-fuzzy controller (the number of coefficients to be tuned) and enables the controller to learn without genetic search algorithms. Using traditional, nonlinear AND/OR operators produces negative results, which makes learning the controller more difficult.

The theoretical analysis of the problem was confirmed by the experimental results of learning the controller in the system with a reference model and a known plant model.

The proposed neuro-fuzzy PID controller is a system with three inputs and one output. But linear operators can be applied also in systems with a larger number of inputs and outputs (a system with n outputs can be composed of n systems with one output). The higher complexity of the neuro-fuzzy system, the more advantageous the application of linear operators, because they diminish rapidly the number of coefficients describing the fuzzy model (controller) surface which must be determined by the neuro-fuzzy network in the learning process, whereas nonlinear operators strongly increase this number.

References

Abel D. (1991): *Fuzzy control — An introduction.* — Automatisierungstechnik, Vol.39, No.12, pp.433–438, (in German).

Babuška R. and Verbruggen H.B. (1995): *Identification of composite linear models via fuzzy clustering.* — Proc. 3rd *European Control Conference*, Rome, Italy, pp.1207–1212.

Baldwin J.F. and Martin T.P. (1995): *Fuzzy modelling in an intelligent data browser.* — Proc. of IEEE/FUZZ 95, Japan, pp.1885–1890.

Driankov D., Hellendoorn H. and Reinfrank M. (1993): *An Introduction to Fuzzy Control.* — Berlin: Springer Verlag.

Grant B. and De Bruijn C. (1993): *Generation of fuzzy control rules and membership functions using neural nets.* — Proc. EUFIT '93, Aachen, Germany, pp.227–231.

Horikawa S., Furuhashi T. and Uchikawa Y. (1992): *On fuzzy modelling using fuzzy neural networks with the backpropagation algorithm.* — IEEE Trans. Neural Networks, Vol.3, No.5, pp.801–806.

Korbicz J., Obuchowicz A. and Uciński D. (1994): *Artificial Neural Networks.* — Akademicka Oficyna Wydawnicza PLJ, Warsaw, Poland, (in Polish).

Masuoka R. and Watanabe N. *et al.* (1990): *Neurofuzzy system - fuzzy inference using a structured neural network.* — Proc. Int. Conf. *Fuzzy Logic & Neural Networks*, Iizuka, Japan, pp.173–177.

Piegat A. (1995): *Fuzzy controller.* — Patent application No. P 312057, 22 December, Polish Patent Office, (in Polish).

Piegat A. and Baccari M. (1995): *Shortcomings of the control with fuzzy controller.* — Proc. Int. Conf. *Methods and Models in Automation and Robotics'94*, Międzyzdroje, Poland, pp.759–762.

Piegat A. and Pluciński M. (1995): *Neural model for rotational movement of underwater vehicles.* — Proc. 2nd Int. Conf. *Applications of Computer Systems*, Szczecin, Poland, pp.35–42, (in English).

Piegat A., Jaszczak S. and Pluciński M. (1996): *Self-learning neuro-fuzzy PID controller without simplifications.* — Proc. 3rd Int. Symp. *Methods and Models in Automation and Robotics*, Międzyzdroje, Poland, pp.1195–1200.

Piegat A. (1996): *"Linearized" neuro-fuzzy networks.* — IEEE Trans. Fuzzy Systems, (submitted).

Pfeifer B.M. (1995): *Symbolic analysis and synthesis of fuzzy controllers by transformations between a fuzzy and crisp form* — Automatisierungstechnik, Vol.43, No.11, pp.514–524, (in German).

Preuss H.P. and Tresp V. (1994): *Neuro-Fuzzy.* — Automatisierungstechnische Praxis, Vol.5, No.5, pp.10–24, (in German).

Preuss H.P. and Ockel J. (1995): *Learning method for fuzzy controller.* — Automatisierungstechnische Praxis, Vol.37, No.7, pp.10–20, (in German).

Takagi H. (1990): *Fusion technology of fuzzy theory and neural networks - survey and future directions* . — Proc. Int. Conf. *Fuzzy Logic & Neural Networks*, Iizuka, Japan, pp.13–26.

Tadeusiewicz R. (1993): *Neural Networks.* — Akademicka Oficyna Wydawnicza RM, Warsaw, Poland, (in Polish).

Wang L.X. (1994): *Adaptive Fuzzy Systems and Control.* — Englewood Cliffs: PTR Prentice Hall.

Zimmerman H.J. (1991): *Fuzzy Set Theory and Its Applications. Second Edition.* — Boston: Kluwer Academic Publishers.

Received: April 4, 1996
Revised: September 1, 1996