Analysis of adaptive neuro-fuzzy PD controller with competitive Petri layers in speed control system for DC motor

Piotr Derugo, Krzysztof Szabat
Wrocław University of Technology
50-372 Wrocław, ul. Smoluchowskiego 19
e-mail: piotr.derugo@pwr.wroc.pl; krzysztof.szabat@pwr.wroc.pl

In the paper the issues related to the application of adaptive neuro-fuzzy controller for speed controller of an electrical motor are considered. Adaptive control structure with reference model (MRAS) is used. The standard controller is modified by the implementation of competitive Petri layers into its internal structure. The proposed modification improves the properties of the drive compared to the control structure with standard neuro-fuzzy controller. Theoretical considerations are confirmed by simulation studies experimental tests done on the laboratory stand.

KEYWORDS: adaptive neuro-fuzzy controller, Peri Layers, competitive layers, MRAS

1. Introduction

The intensive development of novel or modification of existing control structures steams from the continuous increase in requirements for the new equipments [2]. The solutions which are robust to the parameter changes of the plant, energy saving, shorten the transients are look after in industrial application [13], [12], [16]. In the recent years fuzzy control becomes extremely popular in almost all industrial fields including electrical drives. However, the standard fuzzy controllers have some limitation – they are not totally robust to existing disturbances. One of the solution is to connect the fuzzy controllers with adaptive methodology. The examples of such concepts are presented in [10], [11], [3], [5]. Despite the superior performance of such control structure the additional modifications are look after. In the last few years there is a tendency to connect the Petri nets with the fuzzy controller [16], [17]. Despite the classical Petri nets were used in different fields e.g. informatics, manufacturing processes etc. [15], [9], [8], [6] their connection with fuzzy logic improve the adaptive structure characteristic.

The main goal of this paper is to present the advanced concepts for a high-performance drive system with changeable inertia. First a mathematical model of the plant is introduced. Then the adaptive fuzzy controller with Petri nets is described in detail. Next the simulation results are demonstrated and described. After that the results of the experimental tests are shown. The paper is summarized with some concluding remarks.

2. Mathematical model of control system

A typical electrical drive system is composed of a power converter-fed motor coupled to a mechanical system, a microprocessor-based system controller, current speed and/or positions sensors used as feedback signals. Typically, cascade control structure containing two major control loops is used. The inner control loop performs a motor torque regulation and consists of the power converter, electromagnetic part of the motor, current sensor and respective current or torque controller. Therefore, this control loop is designed to provide sufficiently fast torque control, so it can be approximated by an equivalent first order term. If this control is ensured, the driven machine could be AC or DC motor, with no difference in the outer control loop. The outer loop consists of the mechanical part of the motor, speed sensor, speed controller, and is cascaded to the inner loop. It provides speed control according to the reference value. The block diagram of the cascade control structure is shown in Fig. 1 [10].

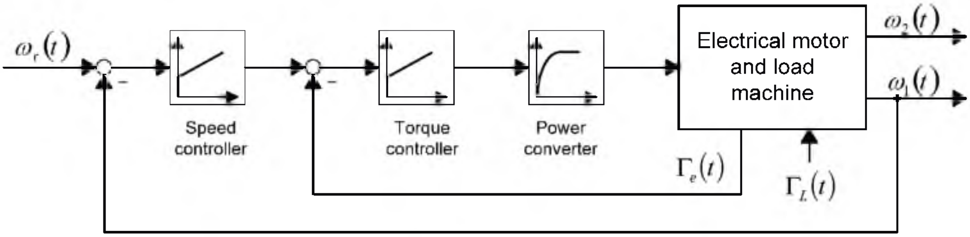

Fig. 1. The classical cascade control structure

In this paper the commonly-used model of the drive system with the resilient coupling is considered. The system is described by the following equations (in per unit system) [3], [7], [14].

$$\dot{\omega}_1 = \frac{1}{T_1}\left(m_e - m_s\right). \tag{1}$$

$$\dot{\omega}_2 = \frac{1}{T_2}\left(m_s - m_L\right). \tag{2}$$

$$\dot{m}_S = \frac{1}{T_C}\left(\omega_1 - \omega_2\right). \tag{3}$$

where: ω_1 – motor speed, ω_2 – load speed, m_e – motor torque, m$_s$ – shaft (torsional) torque, m_L – disturbance torque,T_1– mechanical time constant of the motor, T_2 – mechanical time constant of the load machine, T_c– stiffness time constant.

3. Adaptive neuro-fuzzy controller with competitive Petri layers

The model reference adaptive control structure with the on-line tuned fuzzy controller is proposed for the drive system. The general diagram of the adaptation system is presented in Fig. 2.

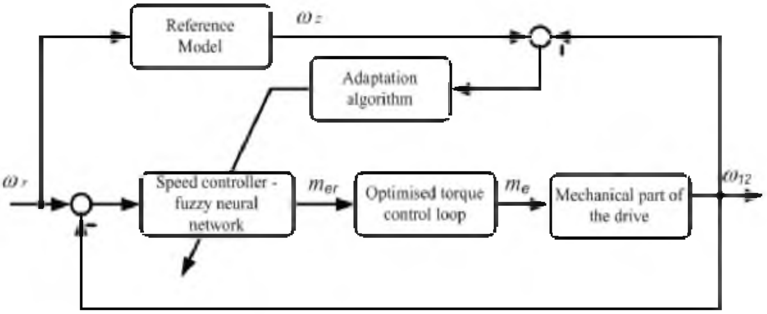
Reference Model
ω_z
Adaptation algorithm
ω_r
Speed controller - fuzzy neural network
m_{er}
Optimised torque control loop
m_e
Mechanical part of the drive
ω_{12}

Fig. 2. Structure of the adaptive control system with the SNFC

The fuzzy controller is tuned so that the actual drive output could follow the output of the reference model. The tracking error is used as the tuning signal. The reference model is chosen as a standard second order term.

$$G_m(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{4}$$

where ζ and ω_n are the assumed damping ratio and resonant frequency.

The supervised gradient descent algorithm is used to tune the parameters $w_1, \ldots, w_M$ of the 4th layer of the neuro-fuzzy structure presented in Fig. 4, to obtain the minimizing of the cost function defined below:

$$J = \frac{1}{2}(\omega_z - \omega_{12})^2 = \frac{1}{2}e_m^2 \tag{5}$$

where e_m – error between model response ω_z and actual speed of the drive system ω_1

Parameter adaptation is obtained using the following expression:

$$w_j(k+1) = w_j(k) + \Delta w_j \tag{6}$$

where:

$$\Delta w_j = -\gamma\frac{\partial J}{\partial w_j} = \gamma\left(-\frac{\partial J}{\partial y_o}\right)\left[\frac{\partial y_o}{\partial w_j}\right] = \gamma\delta_o u_j \tag{7}$$

u_j – is the normalized firing strength of j-th rule, γ – learning rate and

$$\delta_0 = -\frac{\partial J}{\partial y_o} = -\frac{\partial J}{\partial e_m}\frac{\partial e_m}{\partial y_o} = -\frac{\partial J}{\partial e_m}\frac{\partial e_m}{\partial \omega_{12}}\frac{\partial \omega_{12}}{\partial y_o} \tag{8}$$

Expression (9) involves computation of the gradient of ω_1 with respect to the output of the controller y_o, which is the reference electromagnetic torque m_{er}. The exact calculation of this gradient cannot be determined due to the uncertainty of the plant and nonlinear friction characteristic. However, it can be assumed that the change of the drive speed with respect to the motor torque or current is a monotonic increasing process. Thus, this gradient can be approximated by some positive constant values. Owing to the nature of the gradient descent search only the sign of the gradient is critical to the iterative algorithm convergence. So the adaptation law of the controller parameters can be written as:

$$w_j(k+1) = w_j(k) + \gamma \delta_o u_j \cong w_j(k) + \gamma e_m u_j \tag{9}$$

The learning speed of the above algorithm is usually not satisfactory due to the slow convergence. To overcome this weakness, a modified algorithm based on local gradient PD control is used:

$$\delta_o \cong e_m + \Delta e_m \tag{10}$$

where Δe_m is the derivative of the e_m.

The learning rate γ can be divided into two factors k_p and k_d for e_m and Δe_m, respectively. The derivative term is used to suppress a large gradient rate. Thus, in formula (10), the similarity to the back propagation algorithm (with the learning rate and momentum factor) used for neural network training can be seen.

The general diagram of the used fuzzy controller is presented in Fig. 3. It describes the relationship between the speed error *e(k),* its change *Δe(k)* and change of the control signal *Δu(k).* [12], [1], [18],

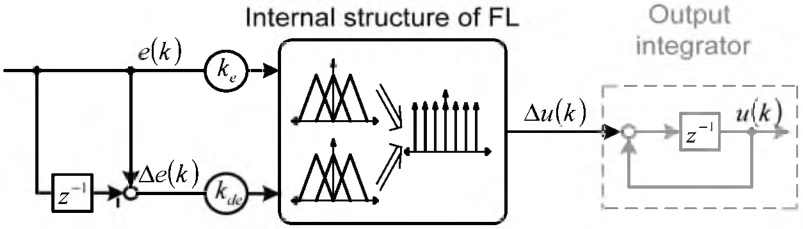

Fig. 3. A general structure of a fuzzy controller

The PI-type fuzzy logic system has an output integrator presented in Fig. 2 with the dotted line. On the contrary, the PD-type structure consider dint the paper does not have it. The control surface of the fuzzy system can be also seen as the nonlinear switching function. So this controller is treated as the fuzzy sliding-mode controller (FSMC) in the present study.

According to the literature, two main frameworks of the fuzzy sliding mode control can be distinguished. In the first one the switching surface *s* is calculated directly. Then, on the basis of the obtained function *s*, the nonlinear switching function is approximated by the fuzzy system. In the second framework the switching surface is not directly visible. It can be calculated from the properties of the rule base, which describe the relationship between the error (e) and the change of the error (Δe). Similarly to the classical sliding mode control, the switching surface is described by:

$$s^* = \lambda^* e^* + \Delta e^* \tag{11}$$

where λ^* represents the slope of the switching line $\lambda^* e^* + \Delta e^* = 0$. The parameter λ^* can be calculated based on the range of the universe of discourse of the input variable (e, Δe).

The rule base of the fuzzy controller incorporates several IF-THEN rules. They can be written in the following form:

$$R_j\text{: } IF\ x_1\ is\ A_1^j\ and\ x_2\ is\ A_2^1\ THEN\ y = w_i, \tag{12}$$

where x_i – input variable of the system, A_i^j – input membership function, w_i – consequent function.

Schematic diagram of the adaptive neuro-fuzzy controller with competitive Petri layers is shown in Fig. 4.

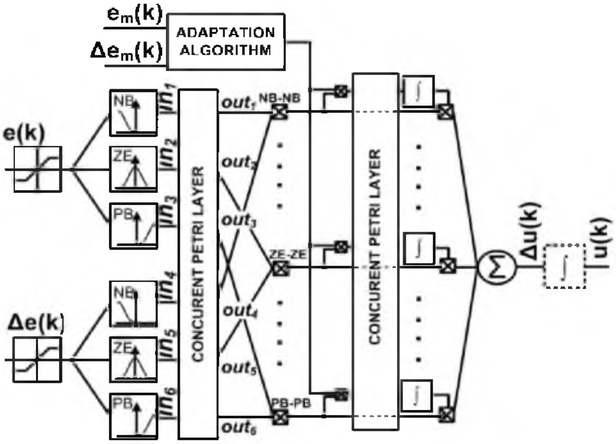

Fig. 4. Adaptive neuro-fuzzy controller with two Petri layers

This system can be divided into several layers with the following functions.

Layer 1. The nodes of the first layer pass the input signal into the second layer. Each node in this layer corresponds to a specific input variables (x_1=$e(k)$; x_2=$\Delta e(k)$).

Layer 2. Each node of this layer represents the specific input membership functions A_i^j. In this point the input signals are fuzzified.

Layer 3. Each node in this layer represents the premises part of fuzzy rule. The symbol Π represents the used *t*-norm operation which in this study is the *prod* function. It means that the incoming signals of each rule are multiplied. Then the result is send to the next layer.

Layer 4. In this layer the output signal of the fuzzy system is calculated. First, the incoming signals are multiplied by the output singletons w_j. Then, according to (4), the defuzzyfication procedure known as the singleton method is performed:

$$\Delta u = y_o = \sum_{j=1}^{M} w_j u_j \tag{13}$$

In this paper, implementation of a competitive Petri layer [16], [17] for the fuzzy controller with 9 rules and Gaussian-type input functions is investigated. Two possible placed of a competitive layer are tested. In the first case the Petri Layer is put between input membership functions and inference layer, which is a solution corresponding to [16], [17]

In the second case, the Petri layer is implemented into adaptation mechanism of the controller. In that solution, an analogy to a competitive learning Winner Takes it All (WTA) [4] (typical for neural networks) can be seen.

Competitive Petri layer specifies the number of the input signals which are transferred to the outputs according to the following formulas:

$$\mathrm{A} = \underset{k=1\ldots N_d \leq n}{\max\_k} \left\{ \underset{i=1\ldots n}{sort}\ \underline{in} \right\} \tag{14}$$

$$\underset{\substack{in_i \in \mathrm{A} \\ i=1\ldots n}}{\forall} out_i = in_i \tag{15}$$

$$\underset{\substack{in_i \notin \mathrm{A} \\ i=1\ldots n}}{\forall} out_i = 0 \tag{16}$$

where A – vector of k maximum values from input vector, *max_k* – selection operator of k maximum values from the input vector, *sort* - operator of sorting vector by absolute values, *in* - input vector, N_d - maximum number of values designed to remain active, n - the length of the input and output vectors.

The considered neuro-fuzzy controller has two inputs with each have three membership functions. It creates in the output the of this layer 6 signals. Accordingly to the structure of neuro-fuzzy system the output of the successive layer consist of 9 signals. Competitive Petri layer acts like input vector critic. It transfers to its output only signals greater or equal to *k-th* smallest signal in terms of its absolute value. The other smaller signals are rejected.

4. Simulation results

In this section the results related to the simulation study of adaptive neuro-fuzzy controller with competitive Petri layers are shown. Two quality indexes are considered. B1 is integer of squared error (ISE) for the first 5 seconds of the simulation. During this time system adapts its weight significantly. Second indicator B2 is integer of squared error from 5th to 15th seconds when weights are changed only slightly. This index is connected to the reaction of the system to the application of the load torque.

Analyzing Tables 1 and 2 it is clearly seen that Petri Layers are giving good effects in case of B1 indicator. Especially layer *k1* ensures good performance of the system. Improvement in B1 indicator means improvement of initial adaptation possibility of the regulator. Unfortunately any other value of k1 coefficient than 0, results in very big increase of B2 indicator.

Table 1. Summary value changes of B1 (0-5 s) control quality indicator

k_2 k_1	0	1	2	3	4	5	6	7	8
0	**0,0%**	-0,3%	-0,5%	-1,5%	-1,1%	-9,4%	**-10,3%**	-8,2%	116%
1	-28,5%	-28,5%	-28,5%	-28,5%	-29,7%	-31,0%	-27,1%	-25,3%	117%
2	-25,7%	-25,7%	-25,7%	-25,7%	-25,7%	-25,7%	**-29,2%**	-25,7%	117%
3	-28,5%	-28,5%	-28,5%	-28,5%	-28,5%	-28,5%	-28,5%	-28,5%	120%
4	205%	205%	205%	205%	205%	205%	205%	205%	205%
5	NS	NS	NS	NS	NS	NS	NS	NS	NS

Table 2. Summary value changes of B2 (5-15s) control quality indicator

k_2 k_1	0	1	2	3	4	5	6	7	8
0	**0,0%**	0,0%	-0,1%	0,0%	0,0%	-0,6%	**-0,6%**	-0,6%	101%
1	486%	486%	486%	486%	478%	466%	460%	479%	92,8%
2	275%	275%	275%	275%	275%	275%	**259%**	276%	93,1%
3	387%	387%	387%	387%	387%	387%	387%	387%	89,7%
4	244%	244%	244%	244%	244%	244%	244%	244%	244%
5	NS	NS	NS	NS	NS	NS	NS	NS	NS

Selected transients of system states are presented in Figures 5-8. Firstly the system without Petri nets is analyzed. The system transients are shown in Fig. 5. As can be seen the initial values of the weight are set to zeros. Then, as the time pass by, their values is changing to optimal position in order to minimize the tracking error of the plant.

Then the system with one Petri layer is tested. The transients of system with implemented Petri Layer that resets 6 signals in k_2 layer are shown in Fig. 6. The biggest difference between the Fig. 5 and 6 evident in the weight transients. The application of the Petri layer results in faster in case of the biggest weights. The disparity between the speeds or torques are much smaller.

Next the system with two Petri layer is examined. In Fig. 7 transients of system with k_2 Petri Layer that resets 6 signals are shown. Significant reduction of difference between measured and reference model velocity in first period of system operation can be seen. During following periods big oscillations in current and bigger errors in speed are evident into transients. What is more oscillations with big frequency and small amplitude appeared in motors velocity.

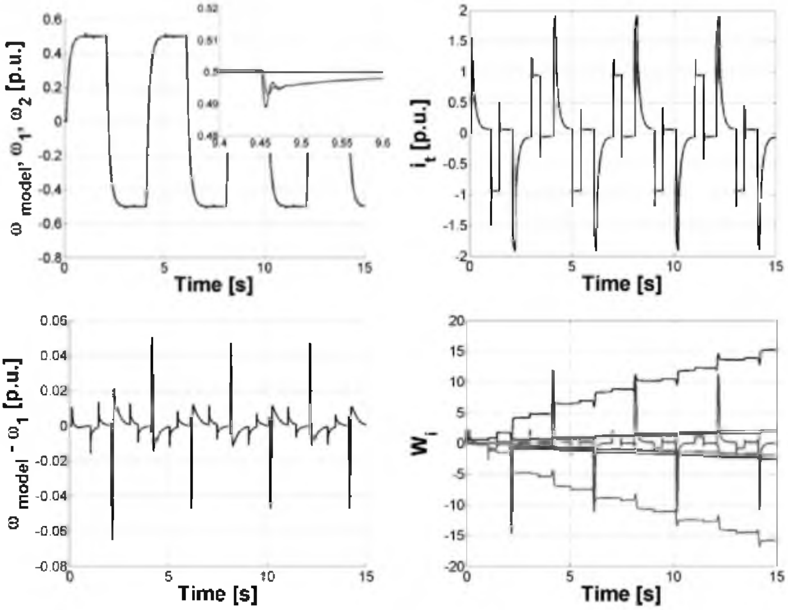

Fig. 5. Transients of system without Petri layers

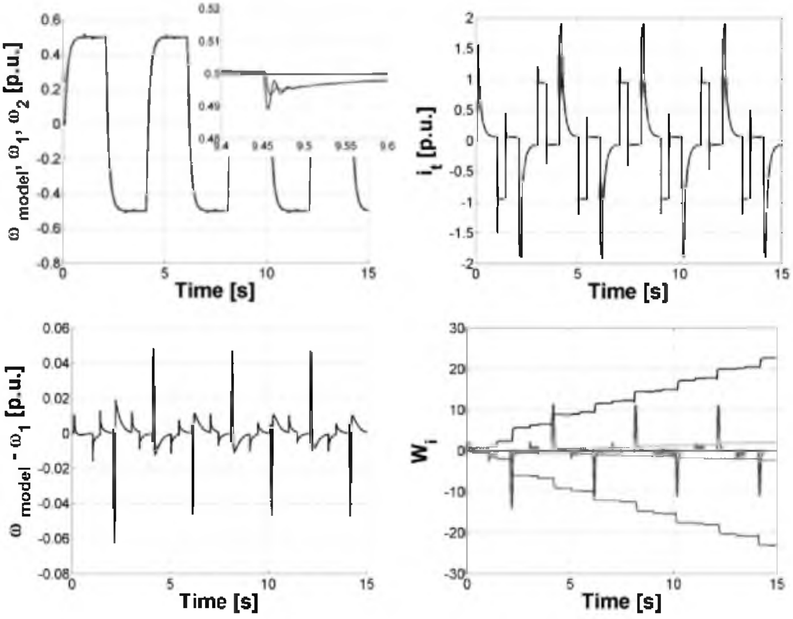

Fig. 6. Transients of system with Petri layers k1 resetting 0 signals and k_2 resetting 6 signals

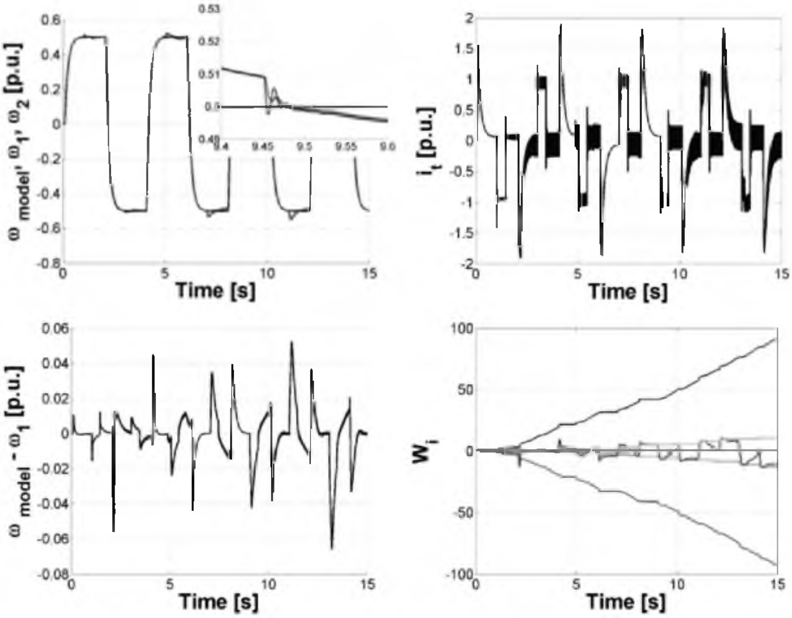

Fig. 7. Transients of system with Petri layers with k_1 resetting 2 signals and k_2 resetting 6 signals

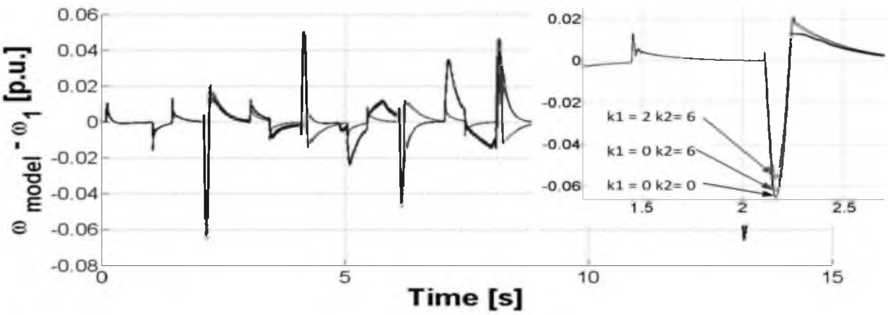

Fig. 8. Comparison of difference between model and motor velocity for different number of resetted signals in Petri Layers

In the Figure 8 transients of differences between reference model and drive velocity for selected systems with and without competitive Petri layers are presented. As can be easily seen, the system with controller with k_2 layer that resets 6 signals allows to obtain better properties of the speed control. The error in dynamic states, such as change of load or reference speed, is smaller than in system with original regulator. Looking at transient of system with two layers, k_1 resetting 2 and k_2 resetting 6 signals, it can be easy to noticed that tracking errors are much smaller for reference changes. However during steady states, especially in later periods of system operation, errors are bigger. This suggests that it is worthwhile to consider switching layer parameters during system operation.

5. Experimental result

Experimental studies have been carried on a laboratory system consisting of two DC motors. Power output is a H bridge. Control is performed by PC computer equipped with digital signal processor and dSPACE 1106 card. Control circuit is sampled at 2 kHz frequency. Two mass system is obtained using elastic shaft. Block diagram of the test system is shown in Figure 9.

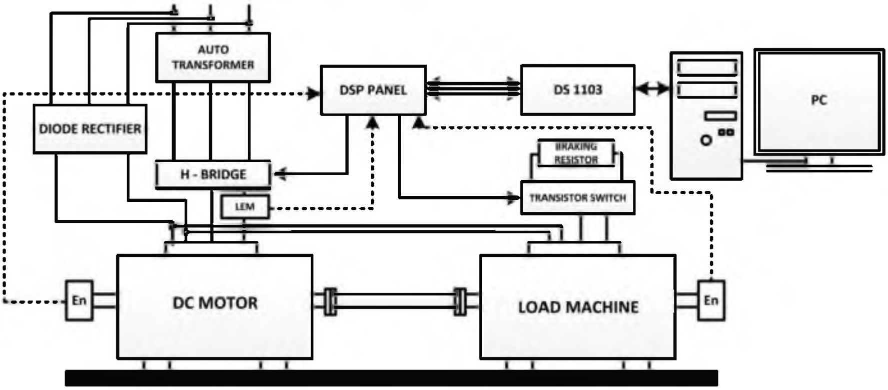

Fig. 9. Block diagram of the experimental

After simulations, for experimental study three cases were selected, original regulator without Petri layers as reference point, regulator with k_2 layer resetting 6 signals and regulator with both layers k_1 resetting 2 signals and k_2 resetting 6 signals. Results are shown in Table 3. Indicators B1 and B2 are defined as in simulations.

As it can be seen both indicators showed similar behavior as during simulations.

Table 3. Summary value changes of B1 and B2 control quality indicators

	$k_1 = 0; k_2 = 0$	$k_1 = 0; k_2 = 6$	$k_1 = 2; k_2 = 6$
B1	0,0%	-0,7%	-26,9%
B2	0,0%	-15,2%	115,3%

Similarly to simulations, in Figures 10-12 some of chosen system transients are presented for same parameters of k_1 and k_2 layers. Comparing them big increase of difference between motor speed and reference model speed in case of system with regulator with 2 layers (Fig. 12) can be seen.

What is also clearly visible rate of weight coefficients in systems with Petri layers is much bigger than in original system without Petri layers.

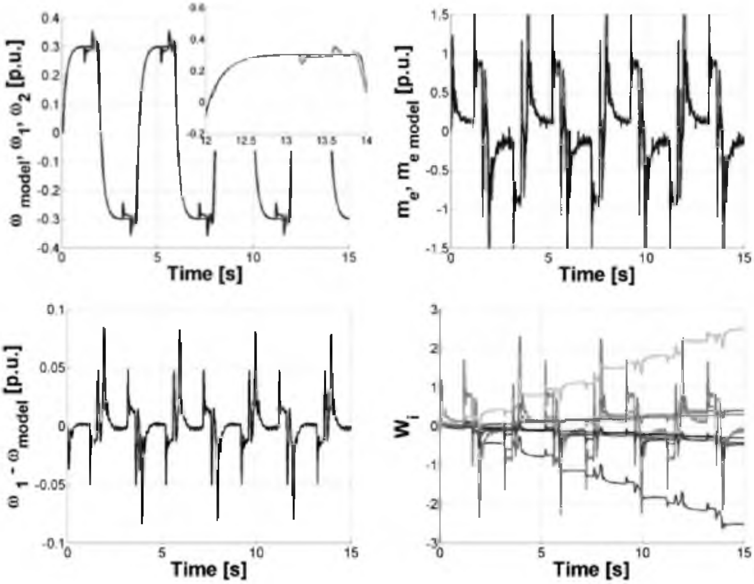

Fig. 10. Chosen experimental transients of system with PD regulator without Petri layers

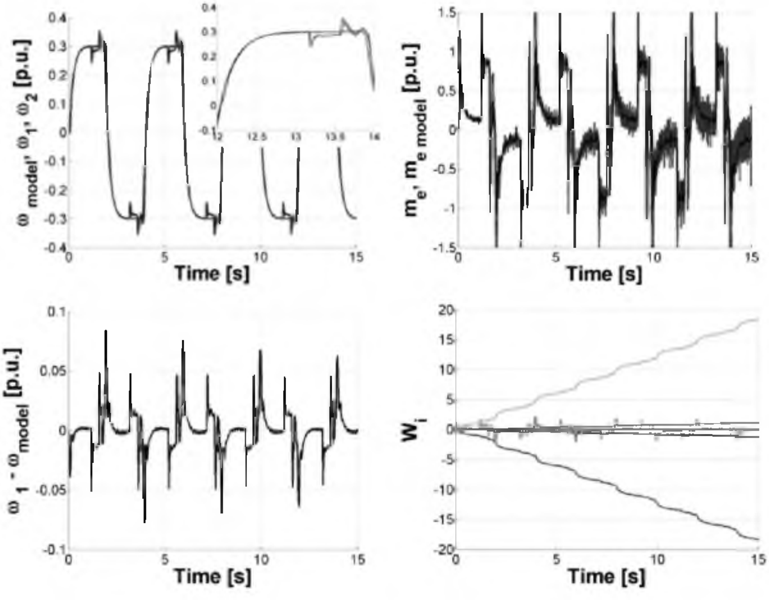

Fig. 11. Chosen experimental transients of system with PD regulator with Petri layers k1 resetting 0 signals and k_2 resetting 6 signals

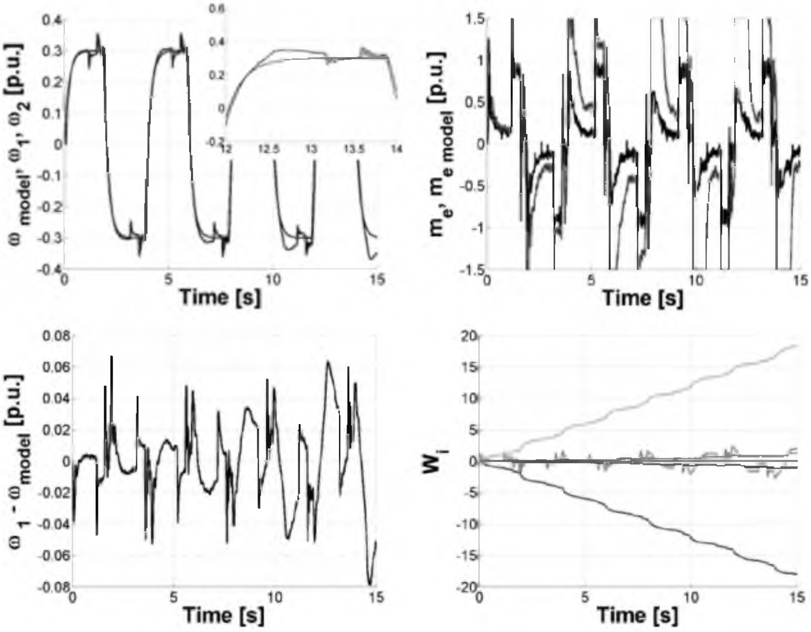

Fig. 12. Chosen experimental transients of system with PD regulator with Petri layers k1 resetting 2 signals and k_2 resetting 6 signals

Figure 13 shows comparison of difference between motor and reference model speed transients for different investigated Petri layers implementations. Similarly as in simulations system with regulator with k2 layer resetting 6 signals smaller errors over the transient can be noticed. Also transient of system with both Petri layers acts as in simulation tests, much smaller errors during motor returns, and much bigger errors in static states especially in later periods of system work.

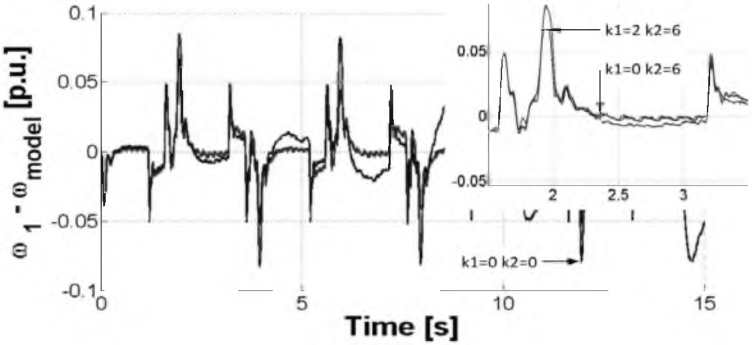

Fig. 13. Comparison of difference between model and motor velocity for different number of resetted signals in Petri Layers

6. Conclusions

In the paper an adaptive control structures based on the MRAS concept have been investigated. As a classical approach the system with standard neuro-fuzzy controller is implemented. The obtained results confirm good dynamic properties of the drive, Then the control structure with the neuro-fuzzy speed controller with Petri layers are considered. This control structure ensures better performance of the drive systems. The tracking error is reduced by 30% in the first 5 second of the work. However, it should be noticed that in next period of the system work the application of the Petri layer is not so effective. During experimental research theory has been proven, system behaved in similar way as in simulations. In the next study the system with variable Petri nets will be considered.

Acknowledgment

This research work is supported by National Science Centre (Poland) under grant Adaptive fuzzy control of the complex drive system with changeable parameters UMO-2011/03/B/ST7/02517 (2012-2015).

References

[1] Brock, S., User-friendly design method for fuzzy speed controller for servo drives, Proceedings of the First Workshop on Robot Motion and Control, 1999. RoMoCo '99., 17 – 21.

[2] Brock, S., Zawirski, K., New approaches to selected problems of precise speed and position control of drives, IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society, 6291 – 6296.

[3] Derugo, P. Dybkowski, M. Szabat, K. , Adaptacyjne wektorowe sterowanie układem napędowym z połączeniem sprężystym, Prace Naukowe Instytutu Maszyn, Napędów i Pomiarów Elektrycznych Politechniki Wrocławskiej. Studia i Materiały Vol. 66, nr 32, t. 2 (2012) 67—76.

[4] Długosz R., Talaśka T., Pedrycz W., Wojtyna R., *Realization of the Conscience Mechanism in CMOS Implementation of Winner-Takes-All Self-Organizing Neural Networks,* IEEE Transactions on Neural Networks, Volume: 21, NO. 6, (2010), 961-971.

[5] Dybkowski, M. Orłowska-Kowalska, T. Kapela, D., Analiza wpływu metody adaptacji regulatora prędkości na właściwości dynamiczne napędu indukcyjnego, Prace Naukowe Instytutu Maszyn, Napędów i Pomiarów Elektrycznych Politechniki Wrocławskiej. Studia i Materiały, Vol. 64, nr 30, (2010), 202-210.

[6] Henry M., Layer R., Snow K., Zaret D., *Evaluating the risk of cyber attacks on scada systems via petri net analysis with application to hazardous liquid loading operations*, Technologies for Homeland Security, (2009), 607–614.

[7] Kamiński, M. Szabat, K., Design of the state controller for the two-mass drive system including the dynamic of the torque control loop, Poznan University of Technology Academic Journals. Electrical Engineering, No. 61, (2010) 89-100.

[8] Lasota A. Karatkevich A., *Modification of Petri nets modeling production processes when quality control is introduced*, Przegląd Elektrotechniczny, (2011) R. 87, nr 11, 164-168.

[9] Łebkowski P., *Sieci Petri narzędziem analizy łańcucha dostaw*, Logistyka, nr 5, (2007), 395-401.

[10] Orlowska-Kowalska T., Szabat K., Jaszczak K., *The influence of parameters and structure of PI-type fuzzy-logic controller on DC drive system dynamics*, Fuzzy Sets and Systems 131 (2002), 251–264.

[11] Orłowska-Kowalska T., Dybkowski M., Szabat K., *Adaptive neuro-fuzzy control of the sensorless induction motor drive system*', 12th International Power Electronics and Motion Control Conference EPE-PEMC (2006), 1836-1841.

[12] Pajchrowski, T. Zawirski, K., Robust Speed Controller of PMSM based on Adaptive Neuro-Fuzzy Inference System, Przegląd Elektrotechniczny, R. 85, nr 8, (2009), 13-17.

[13] Pajchrowski,T., Zawirski, K., Fuzzy Logic Speed Controller Robust Against Drive Parameters Variation, 2006. EPE-PEMC 2006. 12th International Power Electronics and Motion Control Conference, 332 – 336.

[14] Paul, P. Zawirski, K., Odtwarzanie niedostępnych zmiennych stanu napędu z połączeniem sprężystym, Zeszyty Naukowe Wyższej Szkoły Informatyki, Vol. 2, Nr 1, (2003), 69-85.

[15] Petri C.A., *Kommunikation mit Automaten, Schrıften des Rheinisch-Westfalischen Insitutes fur Instrumentelle mathematak an der Universitat Bonn,* Heft 2, Bonn, Germany (1962); translation: C. F. Greene, supplement 1 to Tech. Rep. RADC-TR-65-337, Vol. 1, (1965), 89.

[16] Wai Rong-Jong, Liu Chia-Ming, *Robust Control for Linear Induction Motor Servo Drive Using Neural Network Uncertainty Observer,* IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 56, NO. 7, JULY (2009), 2667-2683.

[17] Wai Rong-Jong, Chu Chia-Chin, *Motion Control of Linear Induction Motor via Petri Fuzzy Neural Network*; IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 54, NO. 1, (2007), 281-295.

[18] Zawirski, K. Urbanski, K., Application of fuzzy control technique to thyristor DC drive, Proceedings of the IEEE International Symposium on Industrial Electronics, 1997. 1173 - 1178 vol.3.