

## APPLICATION OF COMPETITIVE AND TRANSITION PETRI LAYERS IN ADAPTIVE NEURO-FUZZY CONTROLLER\*

PIOTR DERUGO

Wrocław University of Science and Technology, Department of Electrical Machines,  
Drives and Measurements, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland,  
e-mail: piotr.derugo@pwr.edu.pl

**Abstract:** The article is a summary of previous work on the possibility of using Petri layers in adaptive neuro-fuzzy controllers. In the first part of the paper the controller and two types of Petri layer have been presented, competitive layer which resets certain signals and transition layer which causes omission of signals. Layer properties were described and comparison has been made. In the second part of the paper, the results of a simulation showing the advantages and disadvantages of proposed solutions have been presented. Both quality of reference signal tracking and energetic cost of control process have been calculated. In the last part, analysis and comments on the results were made. Main conclusions are that transition Petri layer can significantly reduce growth of numerical cost of the algorithm despite the increase of fuzzy rules count. Also both competitive Petri layer and transition Petri layer by changing some inner signals can affect output value of the fuzzy system and thus the control quality indicators change. Most positive solutions have been pointed out.

**Keywords:** *Petri layer, neuro-fuzzy, MRAS competitive layer, transition layer*

### 1. INTRODUCTION

Many researchers nowadays have focused on improving the quality of regulation through the development of new control algorithms and modification of existing ones. Examples include predictive control [12], adaptive control [8], [14] or fuzzy algorithms [10], [11], [13]. Especially the last experience growing interest. Among others, Petri layers are one of the many currently available modifications of the neuro-fuzzy controllers [15]–[17]. Not only do they allow the control function of the controller to be changed, but also its computational cost to be optimized, which is one of the factors allowing for cheaper implementation of these regulators in industrial systems, or using them for objects characterized by smaller time constants.

---

\* Manuscript received: April 19, 2016; accepted: July 4, 2016.

The purpose of this article is to present the possibilities of implementing Competitive Petri Layer (CPL) and Transition Petri Layer (TPL) in Adaptive Neuro-Fuzzy Controller (ANFC). The CPL changes output control value of the controller by setting some of the signals existing in the neuro-fuzzy inference system to zero, whereas TPL omits some operations in the algorithm instead of performing idle operations such as adding zero. In terms of mathematical operations both TPL and CPL are identical, of course as long as signals set to zero by CPL and omitted in algorithm by TPL are the same.

Despite the seemingly more complicated algorithm – the algorithm needs to determine which membership functions are to be most active during current iteration – the number of operations needed to compute output value of the controller with TPL dumping all but two membership functions for each input, is significantly smaller than it is in the controller without TPL. This is true especially for controllers with a large number of rules.

At the same time, the controllers with TPL with triangular membership functions allow for the same control function as controllers without Petri layer when the number of signals set to zero is less than or equal to the number of zero values existing by definition. By the number of zero values existing by definition, the author means the number of signals having zero value due to the finite support of triangular function.

The paper is constructed in the following manner. The first part describes the aforementioned adaptive neuro-fuzzy controller and each of the layers conventionally found in it [2], [14], as well as all possible CPL implementations considered so far [3]–[7]. Then in the second part, an analysis of the influence of CPL and TPL properties on ANFC algorithm was carried out. The analysis consists of assessing the time needed to compute the algorithm and changes in the quality control indicators of the system with original and modified controller. Changes in the control function, especially in the potential of its shaping towards the controlled object translate into the quality properties of the resulting control system. Reducing the computational cost allows shorter calculation time of a single iteration to be obtained. Finally, a discussion over individual properties with a commentary about the legitimacy of the use of particular solutions in various control systems was held.

## 2. ADAPTIVE NEURO-FUZZY CONTROLLER WITH PETRI LAYERS

This paper presents the two types of Petri layers possible to implement in the adaptive neuro-fuzzy controller [2], [14]. These layers are CPL and TPL. A schematic diagram of the controller with possible Petri layers is presented in Fig. 1.

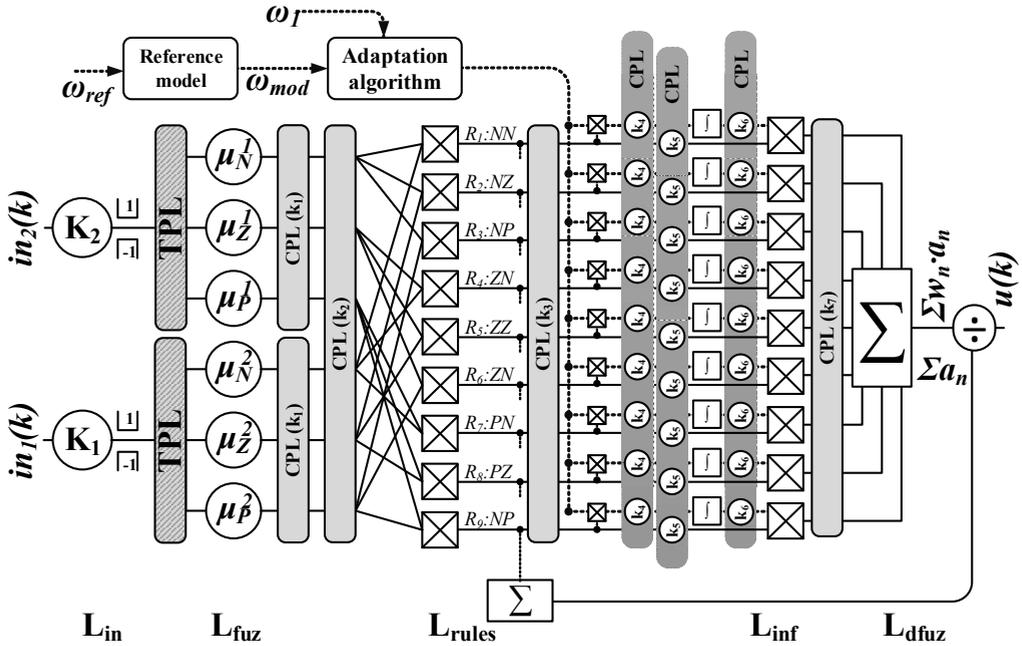


Fig. 1. Schematic diagram of the controller with possible Petri layers

For clarity, a two input ( $in_1$ ,  $in_2$ ) neuro-fuzzy controller has been used, the solution can easily be extended to a larger number if inputs [2]. Input variables can be selected in accordance with the will of the designer. Neuro-fuzzy inference system consists of 6 basic layers [3], [14], with PL implemented among them. The layers of the controller can be described as follows.

## 2.1. INPUT LAYER

The input layer ( $L_{in}$ ), according to formula (1) normalizes input signals. The range of each of the variables is limited to  $\pm 1$ . Scaling factors ( $K_{1,2}$ ) cause the possible range of input values fit the desired limited range of the expected value of the variable

$$L_{in\_out} = \max[\min[(K_n \cdot in_n), 1], -1], \quad (1)$$

where  $n = [e(k), \Delta e(k), \Sigma e(k), u(k-1), x(k), \dots, a]$ .

## 2.2. TRANSITION PETRI LAYER

In the transition layer (TPL), the identification of the range of the input variable, in which the variable is located in the current iteration of the algorithm, is made. This

layer allows the appropriate membership functions and rules based on the identified area of the control plane to be selected. Only weight factors corresponding to the active rules are being adapted. This is dictated by the desire to reduce the numerical complexity of the controller with a large number of rules of the algorithm.

In Fig. 2, an example of control plane and the consequences of TPL application in neuro-fuzzy system with two input and three triangular membership functions for each input is shown. Since only two membership functions (for each input) are active, instead of the four quadrants of control plane, only one needs to be analyzed to obtain all information. This means that during each calculation step only four ( $2^2$ ) out of 9 ( $3^2$ ) rules need to be calculated. Increasing the number of membership functions (for each input) to the  $lr$ , and the number of inputs to  $lw$ , results in the need to determine  $2^{lw}$  instead of  $lr^{lw}$  rules and adapt the same number of weighting coefficients.

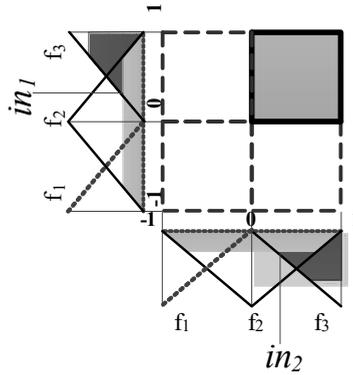


Fig. 2. Control plane for two input controller with triangular membership functions with TPL

Triangular membership functions have a finite medium, for any input value, maximum two adjacent functions only are active in the non-zero level. Application of TPL will cause omission of the least active functions.

### 2.3. FUZZYFICATION LAYER

In the fuzzyfication layer ( $L_{fuz}$ ) according to formula (2), the membership function activation level is calculated for the current input values. Undermentioned formula describes triangular membership function.

$$[L_{fuz\_out}] = \max \left[ \min \left[ \left( \frac{L_{fuz\_in}^i - a_j^i}{b_j^i - a_j^i} \right), \left( \frac{c_j^i - L_{fuz\_in}^i}{c_j^i - b_j^i} \right) \right], 0 \right]_{\substack{i=1\dots n \\ j=1\dots m}} \quad (2)$$

The triangular membership functions used were distributed uniformly over the variable range. Denotations in the formula:  $i$  – the number of input,  $j$  – the number of membership function (for TPL function identified by the layer as one of the most active),  $n$  – number of inputs (here, 2),  $m$  – membership functions corresponding to the sector identified by the TPL (here, Tr1, Tr2) or the number of membership function corresponding to the input for the controller without TPL (here, 3).

#### 2.4. PETRI LAYERS

For the controller with three membership functions per each of the two inputs, the use of either TPL omitting one membership function or CPL  $k_1$  setting one signal to zero makes no difference in terms of output control function. In the case of TPL based on the identification of a sector corresponding to the current value of input signal only two membership functions values are determined for each input. Similarly CPL resetting one of the functions makes a two non-zero signals for each input.

As a result, the input matrix for rules layer ( $L_{rules}$ ) has the size of  $2^2$  for TPL and  $3^2$  for CPL. In general, the effect of CPL can be described by the formulas [5]

$$A = \max_{k=1\dots N_d \leq n} k \{sort(in)\}, \quad (3)$$

$$\forall_{\substack{out_i \in A \\ i=1\dots n}} out_i = in_i, \quad (4)$$

$$\forall_{\substack{in_i \notin A \\ i=1\dots n}} out_i = 0, \quad (5)$$

where:  $A$  – vector of  $k$  maximum values of the input vector,  $\max_k$  – choice operator of  $k$  maximum values from the input vector,  $sort$  – sorting operator (descending in accordance to the absolute value),  $in$  – input vector,  $in_i$  –  $i$ -th value of input vector,  $out_i$  –  $i$ -th value of output vector,  $N_d$  – set number of the maximum values that should remain active,  $n$  – count of the input and output vectors.

#### 2.5. RULES LAYER

The rules layer ( $L_{rules}$ ) determines the levels of activation of individual rules. Rule base consists of expressions like [9]

$$R_n : \text{IF } e(k) \text{ IS } \mu_{j1i1}(e(k)) \text{ AND } \Delta e(k) \text{ IS } \mu_{j2i2}(\Delta e(k)) \text{ THEN } y = a_n.$$

In the work,  $t$ -norm *prod* was used (6)

$$[L_{rules\_out}] = [\mu_j^i \cdot \mu_j^i]_{\substack{i=1\dots n \\ j=1\dots m}}, \quad (6)$$

where  $n$  and  $m$  are, as in the fuzzyfication layer, the number of inputs and the number of membership function for the controller with or without CPL or active membership function in the case of TPL; while  $\mu_{ij} = L_{fuz\_out\ ij}$ .

## 2.6. INFERENCE LAYER

In the inference layer ( $L_{inf}$ ) in accordance with formula (7) firing levels of rules are calculated. The signals of the firing level is then multiplied with corresponding weight coefficient

$$[L_{inf\_out}] = [L_{rules\_out\ j}^i \cdot W_j^i]_{\substack{i=1\dots n \\ j=1\dots m}} \cdot \quad (7)$$

## 2.7. DEFUZZIFICATION LAYER

The defuzzyfication layer ( $L_{dfuz}$ ) evaluates the output of the fuzzy system. In the case presented a classical singleton defuzzyfication algorithm was used, the algorithm is described by the formula [14], [15]:

$$u(k) = \sum_{\substack{i=1\dots n \\ j=1\dots m}} L_{inf\_out\ j}^i \Big/ \sum_{\substack{i=1\dots n \\ j=1\dots m}} L_{rules\_out\ j}^i \cdot \quad (8)$$

## 2.8. ADAPTATION ALGORITHM

The value of change of weighting coefficients in each iteration is described by the equation [15]–[17]:

$$w_j^i(k+1) = w_j^i(k) + L_{rules\_out\ j}^i(k) \cdot [k_{em} e_m(k) + k_{\Delta m} \Delta e_m(k)] \quad (9)$$

where  $e_m$  is the reference model tracking error, calculated as difference between reference model signal, and driving motor speed ( $e_m = \omega_{mod} - \omega_1$ ). As reference model an inertial second order object was used (10), the parameters being chosen as  $\omega_r = 30$ ,  $\xi = 1$

$$G(s) = \omega_r^2 / (s^2 + 2 \cdot \xi \cdot \omega_r \cdot s + \omega_r^2). \quad (10)$$

In each iteration of the algorithm, each of the adapted weighting coefficients is changed by a value dependent on the current tracking error of the model. Only weight coefficients corresponding to active rules are adapted proportionally to the level of corresponding rule activation. It is important to limit the maximum values of the weighting coefficients and/or to incorporate limit on the controller output, for exam-

ple, by stopping the process of adaptation. Weight coefficients should be limited in such a way as to allow the maximum desired output signal value to be reached. A detailed description of the control structure with reference model was also presented in [1], [5], [14].

### 3. OBJECT OF THE STUDY

For the study, a simulation model in Matlab–Simulink has been created. A block diagram of the model is presented in Fig. 3. The model takes into account the mechanical time constants of the electric motors  $T_1$  and  $T_2$ . Mechanical part also includes viscous and Coulomb friction of the motor ( $M_{F1}$ ) and load ( $M_{F2}$ ). The flexible element was modeled using two of its parameters, the elasticity constant ( $Kc$ ) and damping coefficient ( $D$ ). Both speeds of drive machine ( $\omega_1$ ) and load machine ( $\omega_2$ ) are being measured. Input signals for the mechanical part of the system are external load torque ( $M_O$ ) and torque of the drive ( $M_e$ ), the torque on the flexible shaft is designated as  $M_s$ . The electrical part involves time constant and gain of the electromagnetic circuit ( $K_t$ ,  $T_e$ ), constant excitation flux ( $\psi_f = \text{const}$ ), and the current gauge. The inverter also was included in the model as inertial block. The inner control loop uses a classic current PI controller. The outer speed control loop uses the proposed adaptive neuro-fuzzy controller with various cases of Petri layers.

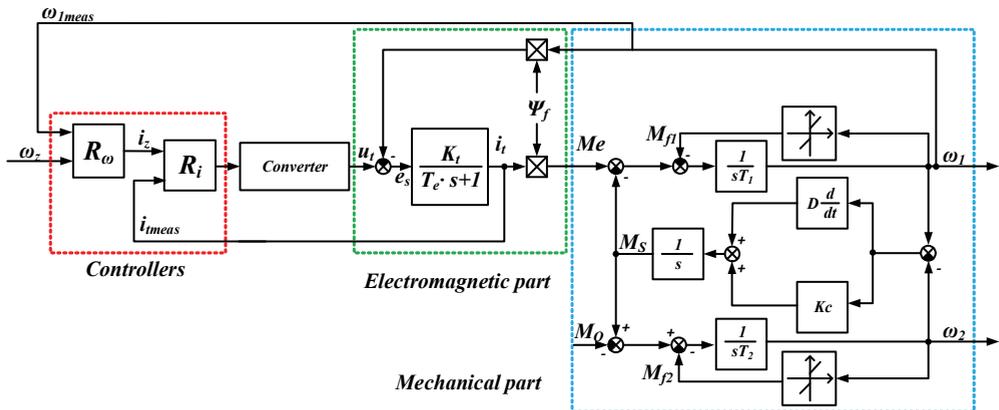


Fig. 3. Block diagram of simulation model

### 4. OUTCOMES

Changes in the control quality criteria depending on implemented Petri layers are gathered in Table 1. White color indicates a reference value of a given criterion, which



off. The 3rd trajectory is the case with constant reference speed, but variable external load torque. The 4th trajectory is analogous to the 2nd trajectory it consists of a dynamic periodically switched load and reversing reference speed trajectory, but in this case, acceleration is performed along the ramp rather than step as in the case of the first trajectory and the second trajectory. The 5th trajectory consists of sinusoidal reference speed with frequency of 0.25 Hz. Trajectory 6 like 5 is sinusoidal but with periodically variable load as in cases 1–3.

Due to the use of the triangular membership function in many cases the output value is identical to the value of the base controller without PL. In the case of triangular functions maximum two functions for each of the inputs are active simultaneously at a non-zero level. Activation levels of other functions are nil. If the number of signals being reset by CPL is the same or less than the number of signals that are zero at the moment, the resulting output value is not changed. In layer  $k_2$  there are six signals, and it was decided not to reset more than two of them to ensure that at least one signal corresponding to each of the inputs will not be zero, which is important from the point of view of the regulation. As can be seen, the best results are obtained using layers  $k_1$ ,  $k_3$  and  $k_5$ , almost all of the observed cases showing improved control quality.

The level of improvement of the proposed control quality indicators varies from less than 1% to 33% for the ISE ( $\omega_1$ ) for trajectory 2 and layer  $k_5$  that reset 8 signals, on the other hand, it causes higher values of energy indicators. However, there are cases, such as  $k_5$  layer resetting 5 signals under trajectory 5 where all of the proposed indicators are lowered (but by less than 1%). The best results in terms of improving the proposed indicators are obtained in the case of layers  $k_1$ ,  $k_3$ , and  $k_5$ .

In Figs. 4–6 the waveforms of selected system with the controller incorporating  $k_1$  CPL and the 2nd trajectory transients are presented. From the point of view of the controller output we can either implement CPL  $k_1$  resetting one of the three signals (Fig. 5) or TPL omitting one signal. In Figs 4–6a, b, c, the motor speed ( $\omega_1$ ), the load speed ( $\omega_2$ ) and the model speed ( $\omega_{mod}$ ) are shown. Figures d show the transient response of the reference model of tracking error, figures c torsional torque ( $M_S$ ), and figures f show reference ( $I_{zad}$ ) and measured current ( $I_{meas}$ ) of armature. The transient in figures g shows the difference between the motor speed and load. In figures h, the waveforms of weight coefficients ( $W_i$ ) are presented.

Analyzing Figs. 4–6 it can be confirmed that a reset or omission of only one signal is identical with the absence of any PL. This is also confirmed by the results in Table 1, and is consistent with theory. The finite carrier of triangular function makes that maximum two membership functions active at the time. Hence the application of TPL omitting one out of the three functions in the controller with triangular functions does not affect the output value and further the quality of control does not change.

Comparing Figs. 4 and 6 it can be seen how PL affects control process. It should be noted that the reference system (without PL) provides a very good reference model

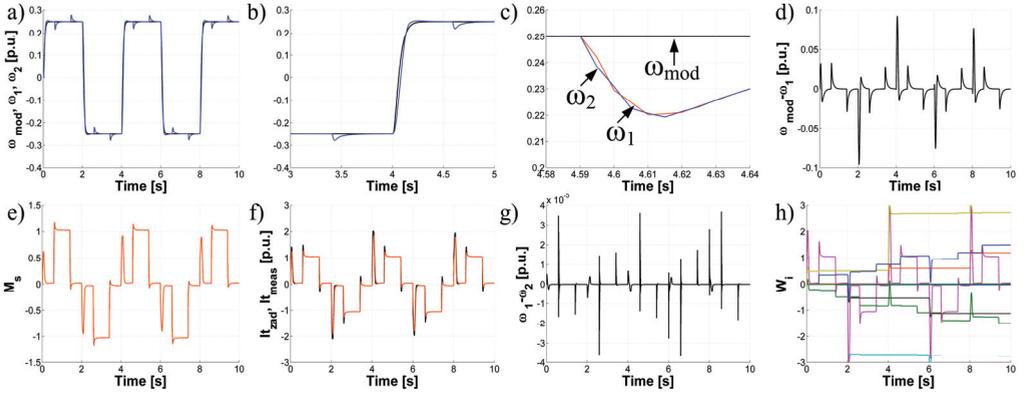


Fig. 4. Transients of a system with the controller without Petri Layer

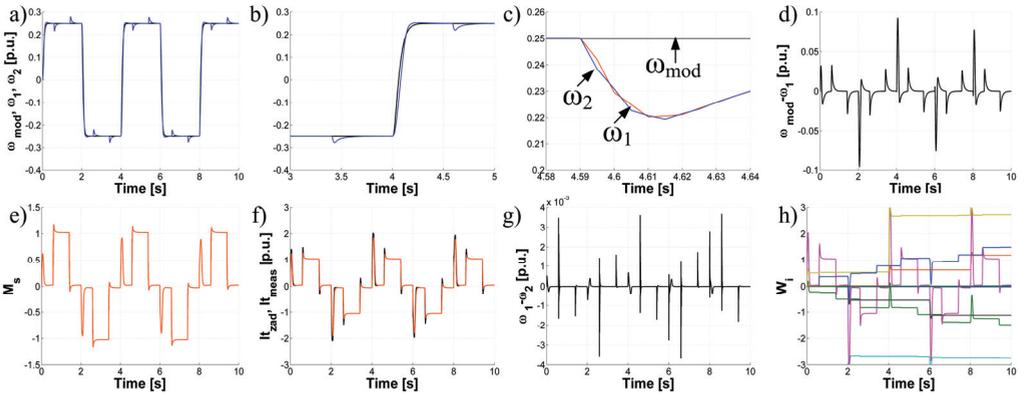


Fig. 5. Transients of a system with a Controller with a  $k_1$  competitive Petri layer resetting one signal/transition layer

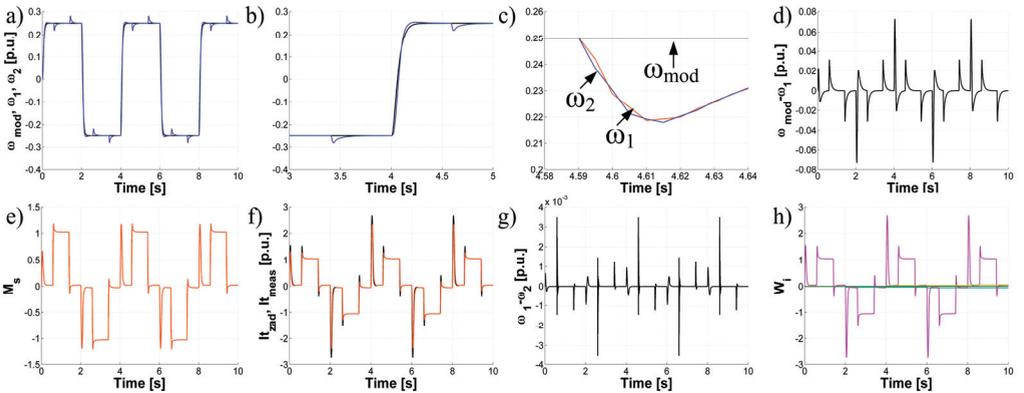


Fig. 6. Transients of a system with a Controller with a  $k_1$  competitive Petri layer resetting two signal/transition layer

tracking quality and the damping of torsional vibration in a dual mass system. Analyzing model tracking error transients (Fig. 4–6 d) it can be seen that in the case of CPL  $k_1$  which resets two signals (Fig. 6) maximum amplitude is reduced. Also, if the differences between the machine and load speed, after initial weight coefficient adaptation, are smaller, especially at moments of reversion ( $t = 6, 8$  s), it can be stated that such a system dampens torsional vibrations better.

Figure 7 presents the execution time of 10 series of simulations with a fundamental step size of 0.0005 s lasting 5 seconds each, that is total of  $10^5$  iterations of the algorithm. The calculations were performed for the cases of the two (PD) and three input (PID) controllers with a different number of membership functions for each of the inputs. As can be seen, the use of TPL significantly reduces the execution time of the proposed test series. Higher order controller (PID) with TPL offers shorter calculation time than the lower order (PD) controller without TPL.

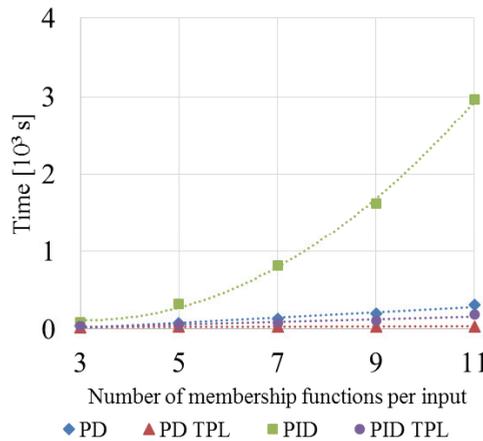


Fig. 7. Execution time of  $10^5$  iterations of the algorithm as a function of the number of membership functions for input

## 5. SUMMARY

The use of CPL allows the controller output value to be changed, which affects the control quality. In particular, the layer  $k_3$  that resets 6 signals and  $k_5$  that resets 6, 7 signals offer significant reduction of all the speed tracking quality indicators proposed.

CPL parameter  $k_1 = 1$  in the case of triangular function provides a control function identical with TPL and the controller without PL. There is no change in output value and thus in quality.

The use of TPL can significantly reduce the numerical cost of the algorithm allowing for the implementation on a cheaper processor or in control loop with shorter

execution time, where the calculation time is critical. Even though the algorithm is more complicated, in the case of controller with 5 or more membership functions per input usage of TPL omitting 3 out of 5 signals reduces the number of necessary mathematical operations and time needed to compute the algorithm is shorter (Fig. 7).

It is possible to implement TPL among other layers than between input and fuzzyfication layers, however, the gained efficiency of calculation time would be smaller, yet it would allow the quality of control of the system to be changed.

The use of PL in ANFC allows the quality control indicators to be improved. Depending on the desired effect, selection of the corresponding layer type and its parameters should be made. It is not always to improve all quality indicators at a time, yet it needs to be stated that after implementation of PL the system was not tuned additionally. The tuning of all controller parameters has been made for the controller without any PL.

#### REFERENCES

- [1] CHEN Y., TENG CH., *A model reference control structure using a fuzzy neural network*, Fuzzy Sets and Systems, 1995, Vol. 73, No. 3, 291–312.
- [2] DERUGO P., SZABAT K., *Adaptive neuro-fuzzy PID controller for nonlinear drive system*, COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, 2015, Vol. 34, No. 3, 792–807.
- [3] DERUGO P., SZABAT K., *A novel implementation algorithm for a fuzzy controller based on the matrix form of the controller*, Przegląd Elektrotechniczny, 2014, Vol. 90, No. 11, 235–238, (in Polish).
- [4] DERUGO P., SZABAT K., *Implementation of the low computational cost fuzzy PID controller for two-mass drive system*, 16th Int. IEEE Conf. and Exposition Power Electronics and Motion Control PEMC, 2014, 564–568.
- [5] DERUGO P., DYBKOWSKI M., SZABAT K., *Analysis of the impact of the position of competitive Petri layer in neuro-fuzzy adaptive controller on the dynamic properties of the drive system*, Przegląd Elektrotechniczny, 2014, Vol. 90, No. 6, 35–39 (in Polish).
- [6] DERUGO P., *Analysis of competitive Petri layers impact on fuzzy Mamdani type regulator performance*, Scientific Papers of the Institute of Electrical Machines, Drives and Measurements of the Wrocław University of Technology. Studies and Research, 2013, Vol. 33, 97–110, (in Polish).
- [7] DERUGO P., DYBKOWSKI M., SZABAT K., *Application of the adaptive neuro-fuzzy speed controller with Petri layers to electrical drives*, Przegląd Elektrotechniczny, 2013, Vol. 89, 64–67.
- [8] JUANG C.F., HSU C.H., *Temperature Control by Chip-Implemented Adaptive Recurrent Fuzzy Controller Designed by Evolutionary Algorithm*, IEEE Trans. Circuits and Systems, 2005, Vol. 52, No. 11, 2376–2384.
- [9] KABZIŃSKI J., KACERKA J., *TSK Fuzzy Modeling with Nonlinear Consequences*, Artificial Intelligence Applications and Innovations, 2014, 498–507.
- [10] KLUSKA J., HAJDUK Z., *Digital Implementation of Fuzzy Petri Net Based on Asynchronous Fuzzy RS Flip-Flop*, Proc. 7th Int. Conf., Zakopane, Poland, 2004.
- [11] KNYCHAS S., SZABAT K., *Adaptive Recurrent Neuro-Fuzzy Control of the Complex Drive System*, EuroCon'2013, Zagreb, Croatia, 2013, 1932–1936.
- [12] LU C.-H., TSAI C.-C., *Generalized predictive control using recurrent fuzzy neural networks for industrial processes*, Journal of Process Control, 2007, Vol. 17, No. 1, 83–92.

- [13] LIN F.J., WAI R.J., LEE C.C., *Fuzzy neural network position controller for ultrasonic motor drive using push-pull DC-DC converter*, IEEE Proc. Control Theory and Applications, 1999, Vol. 146, No. 1, 99–107.
- [14] ORŁOWSKA-KOWALSKA T, SZABAT K., *Control of the drive system with stiff and elastic couplings using adaptive neuro-fuzzy approach*, IEEE Trans. Industrial Electronics, 2007, Vol. 54, No.1, 228–240.
- [15] WAI R., LIU C.H., *Design of Dynamic Petri Recurrent Fuzzy Neural Network and Its Application to Path-Tracking Control of Nonholonomic Mobile Robot*, IEEE Trans. Industrial Electronics, 2009, Vol. 56, No. 7, 2667–2683.
- [16] WAI R., LIU C.H., *Experimental Verification of Dynamic Petri Recurrent-Fuzzy-Neural-Network Path Tracking Control for Mobile Robot*, Int. Conf. on Control and Automation, 2009, 1359–1364.
- [17] WAI R., CHU C.H., *Motion Control of Linear Induction Motor via Petri Fuzzy Neural Network*, IEEE Trans. Industrial Electronics, 2007, Vol. 54, No. 1, 281–295.