Experimental verification of the Takagi-Sugeno fuzzy logic PI controller in stabilization of angular velocity of a DC motor subject to irregular loading

Kunikowski W.[1], Olejnik P.[2*], and Awrejcewicz J.[3]

Abstract: Fuzzy logic control algorithms are regarded to as a relatively new concepts in modern control theory. This paper presents a comparative analysis of two qualitatively different approaches used for angular velocity control of a DC motor subject to chaotic disturbances coming from a gear with a transmission belt carrying a vibrating load. The purpose is to achieve an accurate control of speed of the DC motor (a plant), especially, when the motor parameters and some external loading conditions are partially unknown. First, the classical approach based on the PID control is considered, and then, a fuzzy logic based alternative is proposed. Two different controllers are developed for the purpose of completion of this paper: the classical PID controller and a Takagi-Sugeno type fuzzy logic PI controller. Both control algorithms are implemented on an 8-bit AVR ATmega644PA microcontroller. On the basis of step responses of the plant an analysis as well as an interesting comparison of the controllers' performance has been presented.

Keywords: fuzzy logic controller, PID control, Takagi-Sugeno, DC motor, PWM control, AVR, ATmega, discontinuous system, stick-slip friction.

1. Introduction

Fuzzy logic is a mathematical concept striving to imitate human perception. Instead of numerical values linguistic descriptions are used to characterize input and output variables. The control strategy is derived from expert knowledge and stored in a base of fuzzy rules. This enables the design engineer to describe the behaviour of the object under control with the use of words (linguistic variables), rather than with the use of complex mathematical expressions. Another advantage of the intelligent approach in control of uncertain dynamical systems is the ability of fuzzy logic controllers to handle the control by means of incomplete portion of information. Fuzzy logic has found use in a wide spread of applications like process control, electrical engineering, information technology, image recognition, telecommunications, banking [1,2,3,4].

In paper [5], a PI fuzzy force controller was applied to regulate movement of an industrial robot's end-effector. After generating the robot program, a "foreign" object was introduced to the environment. Without any obstruction during its operation, the robot moves along its preprogrammed path. However, when a contact with the "foreign" object along the way of motion appears, the force control system controls the robot by adjusting the end-effector's position. The force control ensures that the contact forces and moments converge to a desired value. Comparing to the classical PI algorithm, a smaller overshoot and average constant force were achieved.

This work has been supported by the Polish National Science Centre, MAESTRO 2, No. 2012/04/A/ST8/00738.

Kunikowski W. is with Department of Automation, Biomechanics and Mechatronics, Lodz University of Technology, 1/15 Stefanowski Street, 90-924 Lodz, Poland, e-mail: wojciech.kunikowski@dokt.p.lodz.pl

Olejnik P. is with Department of Automation, Biomechanics and Mechatronics, Lodz University of Technology, 1/15 Stefanowski Street, 90-924 Lodz, Poland, e-mail: pawel.olejnik@p.lodz.pl

Awrejcewicz J. is with Department of Automation, Biomechanics and Mechatronics, Lodz University of Technology, 1/15 Stefanowski Street, 90-924 Lodz, Poland, e-mail: jan.awrejcewicz@p.lodz.pl

In [6] a robust fuzzy controller's application has been described for the permanent magnet DC motor. The system parameters concerning the load and the DC motor's constants were unknown. The algorithm was implemented on dSPACE rapid prototyping controller board and connected to Matlab/Simulink programming environment. The fuzzy logic controller has two inputs, a voltage error and its derivative. The control action generated by the controller is the actual voltage supplying the DC motor. It was concluded from the experimental data that the elaborated fuzzy logic controller has achieved shorter response times on pulse input signals and smaller oscillations about the setpoint than the adequate PID controller.

Paper [7] describes an application of a fuzzy controller in Internet traffic management. Web servers suffer from extremely varying load parameters. Sometimes very lightly loaded, sometimes suffering from enormous connection requests. Designing web servers for peak load is not profitable, because even most efficient web servers may still be overloaded by the ever growing population of Internet users. During an overload period, not all users can receive services in timely manner without latency, but on the other hand it is possible to provide a faster connection for premium users. Due to the nonlinear properties of web servers and difficulties in constructing the accurate mathematical model a fuzzy PI controller was proposed. Incoming connections were divided into two classes: premium and basic. The task of regarded fuzzy controller is to maintain a delay ratio between these two classes by assigning suitable number of processes to handle incoming requests. Experimental data shows that significantly lower oscillation and shorter settling time of the delay ratio was achieved comparing to the classical PI controller. Authors state that fuzzy controller improved control quality by approximately 35%.

A hybrid solution of a fuzzy logic and a PI controller was presented in [8]. The object under control consists of a DC motor with a metal disc mounted on its shaft. The load is generated by applying a magnetic field to the disc. The fuzzy logic component is responsible for calculating a gain coefficient for the PI controller. The fuzzy part has three inputs: the reference speed, measured speed and the control action of whole controller from the previous time period. The base of fuzzy rules consists of only one rule, and its aim is to reduce the output of the PI controller in low setpoint speeds. This is due to the nonlinear behavior caused by frictional effects in the mentioned region. This friction compensation method yields faster response of the system and smaller settling time.

An application of fuzzy logic controller (FLC) in electrical engineering was presented in [9]. The controlled object is a self-excited induction generator. The control system was given a task to maintain the steady level of output voltage. Mamdani and Takagi-Sugeno type fuzzy logic PI controllers were developed in the discussed work. Both controllers have two inputs (a voltage error and its derivative) and one output. Performance of proposed solutions was tested against a classical PI controller. Both controllers were programmed in Matlab/Simulink and used to control a model of the self-excited induction generator. Due to high processing power required by the Mamdani controller, the Takagi-Sugeno type FLC was solely implemented on a dSPACE real-time system. It has been concluded from the simulation and experimental data that the fuzzy logic controllers offer significantly better performance compared to the optimally tuned PI type controller, in terms of the response time, settling time and robustness. The downside of the described application is reflected in an increase of the computational performance requirements, especially for the Mamdani type FLC.

In [10] a self-learning Takagi-Sugeno controller was used for identification purposes. Mentioned controller has two input variables with three bell shaped membership functions each. The output of every rule is a linear function of input variables. To implement a gradient-descent learning algorithm, the controller was designed in a form of generalized neural network. The task of the algorithm was to modify the weight coefficients of the neural network to ensure convergence of the output from the fuzzy-neural network to three sets of training data. One of them was obtained from a real object and the other from different mathematical functions. After 200 iterations of the algorithm's execution for each set of data, an average percentage error obtained for the first set of training data was reduced to 1.57%. For the remaining sets the error was reduced to 0.47% and 0.014%, respectively. Although the number of fuzzy rules and input membership functions are preliminarily specified, the algorithm achieved satisfactory results regarding the error and tuning time.

Another example of self-tuning FLC was presented in [11]. A Takagi-Sugeno fuzzy PD controller was applied on a 3-axis milling machine for contour milling. Basing on the position error, the change of the error, the velocity feedback and the fuzzy control action from the previous time period, the output is calculated and sent to the amplifier, which drives the motor. The performance of the controller affects the adaptive algorithm that expands or contracts the input fuzzy sets or even shifts the position of the output sets in the numeric domain. The described FLC was then compared to a well-tuned PID controller. The comparison shows that for cutting straight lines the adaptive fuzzy logic controller achieves two times lower root of mean square contour error and three times lower maximum contour error than the PID controller.

Fuzzy logic controller implemented in an industrial controller was presented in paper [12]. The plant consists of a DC motor subjected to varying load parameters. The regarded Takagi-Sugeno fuzzy logic PI controller was implemented on a PLC controller. A self-tuning fuzzy algorithm which calculates a gain coefficient for the main controller was also implemented. The regarded system's performance was tested for three types of conditions: varying load parameters, varying setpoint velocity and varying setpoint velocity with changing load parameters. In all of these cases the fuzzy logic controller has provided satisfactory results.

In this paper a fuzzy logic PI controller is proposed as an alternative for the classical PID equivalent. Efficiency of both solutions in controlling a multi-degrees-of-freedom discontinuous dynamical system with friction subjected to an irregular external excitation is investigated. In second part of the paper the experimental station and its main components are discussed. Detailed description of both tested numerical algorithms is given in third section. A comparative study of test results of efficiency of the presented fuzzy logic approach is experimentally verified in section four.

2. Experimental station

A model of the dynamical system being a source of instability of the belt driven by the controlled DC motor with a gear (see element 1 in Fig. 4) has been shown in Fig.1. The friction-induced vibration of mass *m* on the moving belt is responsible for the dynamically changing loading of the DC motor's shaft that drives the transmission belt. The non-controlled velocity of belt pulley is very irregular as depicted in Fig. 2.

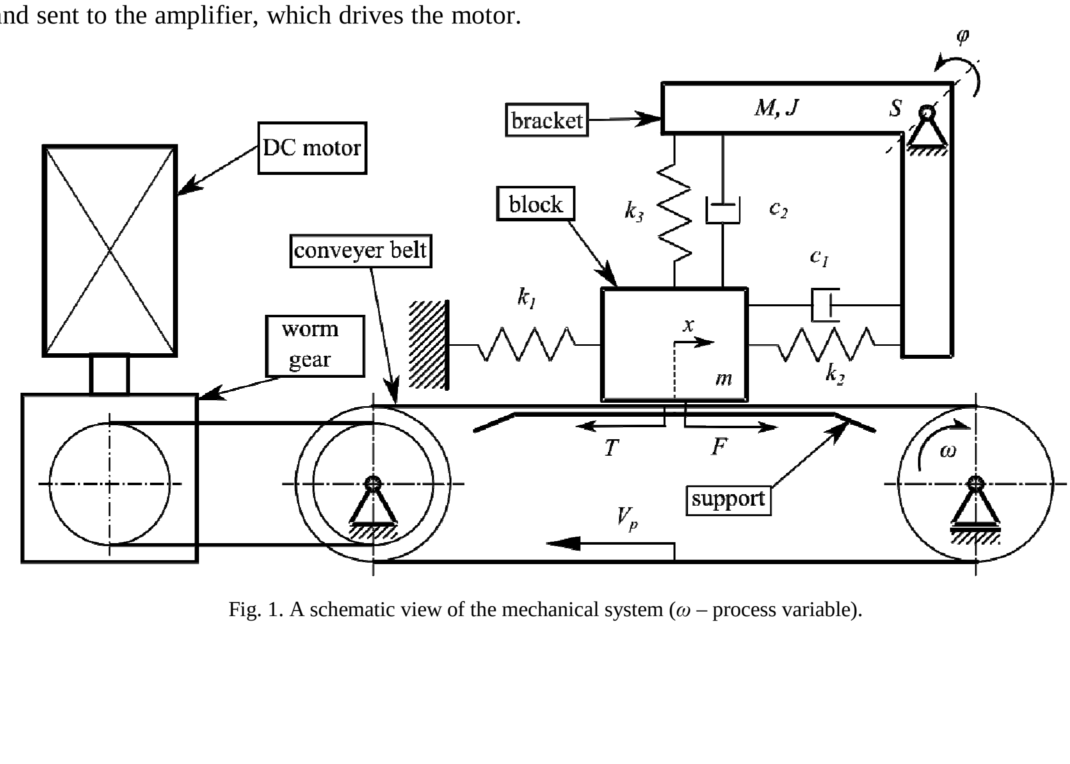

Fig. 1. A schematic view of the mechanical system (ω – process variable).

The controlled object consists of a conveyer belt with a block of mass m oscillating on it along x direction and a bracket of mass M and the mass moment of inertia J , rotating around point S about φ angle. The bracket is attached to the block by means of two linear springs k_2 and k_3 (see Fig. 1). Depending on the linear displacement of the block m and the angular displacement of the bracket J, the friction force in a contact surface between the block and the belt changes rapidly. Moreover, the friction force characteristic switches itself between its kinetic and static form. The experimentally observed irregular changes of frictional force in the block-on-belt model are responsible for the significantly varying load transferred on the DC motor's shaft driving the mechanical system shown in Fig. 1. Constantly changing amplitude of load affects the speed of the gear's conveyer belt. A more detailed description of the model can be found in [13, 14, and 15].

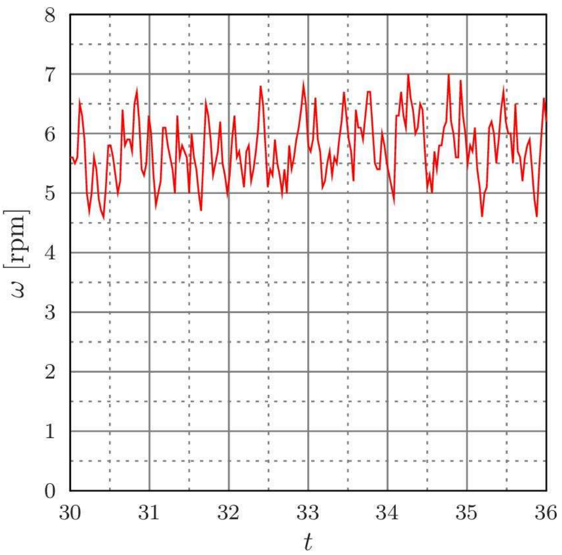

Fig. 2. Speed variations of the uncontrolled system.

Many physical phenomena such as: wear of the worm gear, viscous friction in all of the bearings of the mechanical system, stiffness of the transmission belt, radial run-out of belt pulleys shafts and unevenly distributed value of dry friction coefficient on the surface of transmission belt have a direct impact on stabilization of belt's linear velocity of movement.

To inspect the behaviour of the conveyer belt shown in Fig. 2, the belt pulley's angular velocity was measured by an incremental encoder. Figure 3 depicts exemplary Fourier transform of the system's step response. In that trial the system was controlled by a P controller with the proportional gain K_P = 2.25. The presented spectrum of amplitudes visible in Fig. 3 ends at the Nyquist frequency

$$f_c = \frac{1}{2h} \approx 16.666... \, [Hz] , \quad (1)$$

where: f_c – Nyquist frequency, h = 0.03 s – sampling time.

As can be observed, the system's dynamics is highly irregular. Many variables need to be taken in consideration while designing any optimal controller.

To measure the speed of the transmission belt on which an oscillating mass vibrates, an incremental encoder was used. The 4x encoding was applied for the angular velocity measurement. This means that in a constant time period both rising and falling edges of two shifted measurement lines A and B of the encoder are counted. This method allows to virtually increase the base resolution of the sensor [16]. In this work an 5000 imp/rev encoder was used. Using the 4x encoding and setting the acquisition time to 30 ms yields a measurement error of 0.1 rpm. This means that each subsequent pulse counted with mentioned sampling time increases the output of the velocity recognition algorithm by 0.1 rpm. Unfortunately, the encoder is not perfect. The length of the high state tends to vary due to the manufacturing tolerances. Used method of encoding could magnify that phenomena hence the speed measurement error may be bigger than the assumed 0.1 rpm. The contribution of that error to the shape depicted in Fig. 2 is unknown. No filtering technique was applied to the measurement to capture the dynamics of analysed system as accurate as possible.

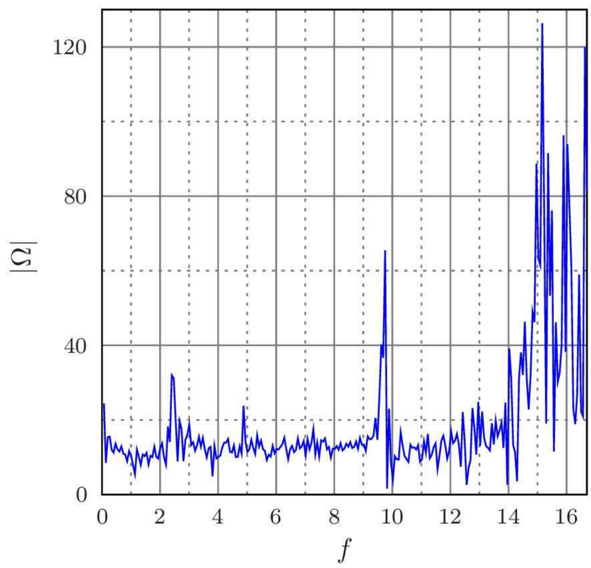

Fig. 3. Fourier transform of the system's step response.

Figure 4 presents the laboratory station on which the experiments have been carried out.

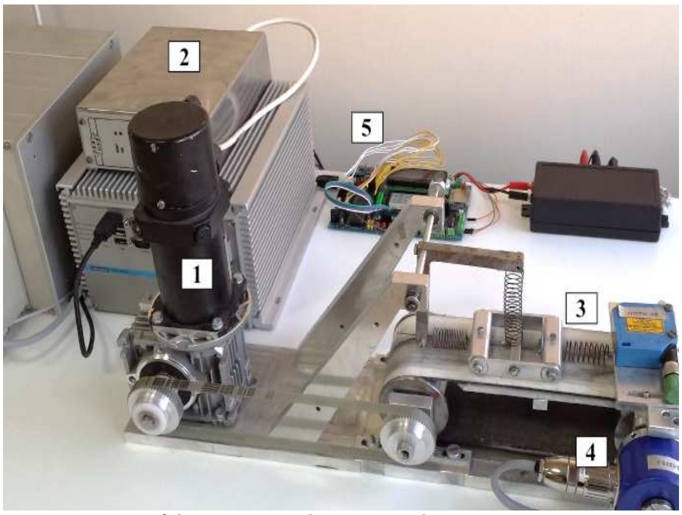

Fig. 4. Picture of the experimental station: 1 - the DC motor PZTK 62-42J with a 15:1 worm gear, 2 - RN12 DC motor driver, 3 - the physical object under control, 4 - IVO GI333 incremental encoder, 5 - Atmel Testing Board 1.03 with ATmega 644PA microcontroller.

Regulation of the angular velocity of the DC motor (1) is made by a control system composed of the Atnel Testing Board (ATB) 1.03 with ATmega 644 PA microcontroller (5) and the RN12 driver (2). Physical system under control (3) influences the speed of the transmission belt measured by the incremental encoder (4). The L293D is powered by 15V power source. The DC motor is powered by the MATRIX 60V (RN12 power circuit) and 30V (RN12 logical circuit) – regulated direct current sources.

Figure 5 presents the ideological wiring diagram of the described system. For the purpose of noise cancelation in the measuring circuit a M74HCT14N Schmitt Inverter was used. It damps voltage peaks causing incorrect number of impulses to be counted by the microcontroller.

To enable compatibility between the microcontroller and the RN12 motor driver a L293D integrated circuit was implemented. The RN12 driver receives a PWM signal with amplitude ranging from 12 to 15V, but the microprocessor is equipped only with 5V TTL outputs. L293D served as an amplifier for the PWM signal. Known parameters of PZTK 62-42J DC motor are as follows: electrical constant: k_e = 0.104 V/rad/s, mechanical constant: k_m= 0.39 N·m/A, armature resistance: R_w = 1.1 Ω armature reluctance: L_w = 0.001 H.

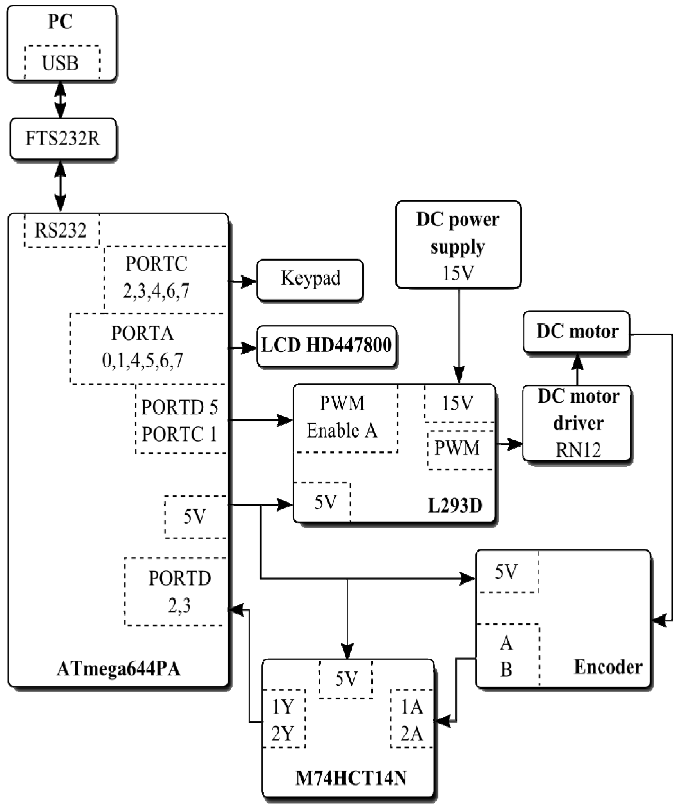

Fig. 5. Wiring diagram of the closed-loop control system

3. Control algorithms

Due to complexity of the control task, a black-box approach to the control problem was used while any mathematical model of the controlled object was omitted. Tuning of the controller was done manually by observing the real time plot of the encoded angular velocity measurement. The goal here was to achieve the lowest possible oscillations of velocity of the belt pulley after reaching the desired setpoint. Both presented algorithms were implemented in C programming language on ATmega644PA microcontroller.

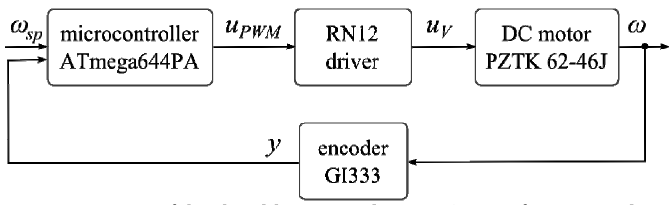

Fig. 6. Diagram of the closed-loop control system (ω_{sp} – reference angular velocity, ω – process variable).

In Fig. 6 the assumed closed-loop control system has been presented. Basing on the setpoint speed (reference angular velocity ω_{sp}) and counted number of impulses y from the encoder, the proposed control algorithm calculates the adequate duty cycle u_{PWM} of the PWM signal. This information is sent to the RN12 driver which generates appropriate voltage u_V for the DC motor. A 10-bit PWM signal with 9.7 kHz frequency was used. Therefore, the variable describing its duty cycle can be changed from 0 to 1023. Control algorithm corrects the PWM duty cycle every 30 ms. The feedback loop time regime is restricted by the accuracy and resolution of the speed measurement.

A. *The PID controller*

Classical discrete PID controller has been presented in Eq. (2).

$$u_{PWM} = K_P e(n) + K_I \sum_{k=0}^{n} e(k) - K_D\left(y(n) - y(n-1)\right), \quad K_I = \frac{T_S}{T_I},\ K_D = \frac{T_D}{T_S}; \tag{2}$$

where: u_{PWM} – output of the regulator, e – error of regulation, K_P – proportional gain, K_I – integral gain, K_D –derivative gain, T_S – sampling time, y – measured value of angular velocity, T_I– integral time coefficient, T_D – derivative time coefficient.

To eliminate rapid responses of the derivative part when the setpoint speed changes (the so called “derivative kick”), the regulator’s variable was changed from the error of regulation to the difference between two successive measurements of speed.

The PID controller’s output was limited to the range [0, 1023] and treated as the new value of the PWM duty cycle.

B. *Fuzzy Logic PI controller*

The second regulator taken into consideration is the Takagi-Sugeno type fuzzy logic PI controller. It has two input values: error e and difference δe between current value of error and value of error from the last cycle. The output value is the increase of PWM duty cycle Δu_{PWM} A classical alternative that would match regulator described earlier is given below

$$\Delta u_{PWM} = K_I e(n) + K_P\left(e(n) - e(n-1)\right), \tag{3}$$

where: Δu_{PWM} – increase in regulator’s output, e – error of regulation, K_P – proportional gain, K_I – integral gain.

Numerical range of each input variable was divided into five triangle and a two piecewise linear L and g fuzzy sets. Output variable was divided into seven one element sets (so called “singletons”). Graphical interpretation of this classification is presented in Fig. 7-9. Formulas for calculating the value of membership function of each type of fuzzy set are given in Eq. (4), (5) and (6). The Takagi-Sugeno controller was chosen due to its defuzzyfication simplicity and lower requirements regarding processing power than the Mamdani type controller.

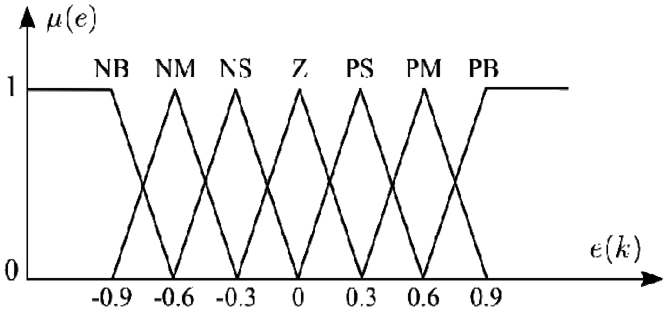

Fig. 7. Fuzzy sets of first input variable – error *e*.

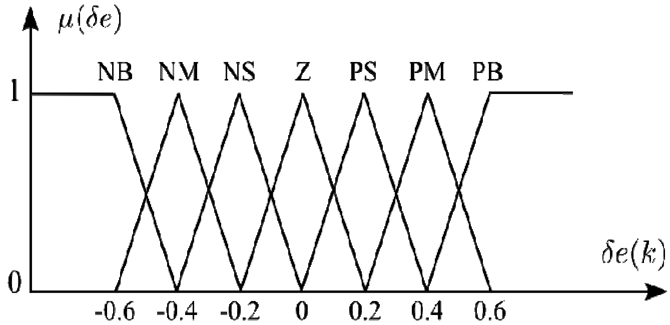

Fig. 8. Fuzzy sets of second input variable – increase of error *δe*.

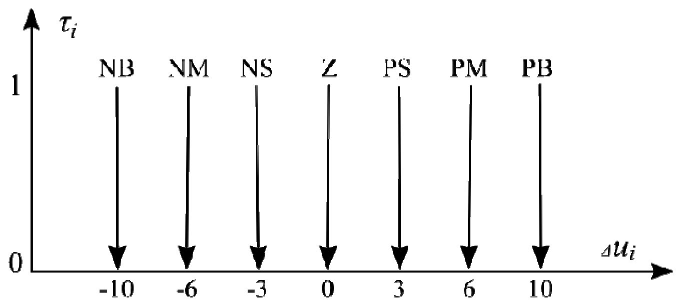

Fig. 9. Fuzzy sets of output variable Δu_{PWM}.

In Fig. 7-9 the thresholds of fuzzy sets are as follows: NB – negative big, NM – negative medium, NS – negative small, Z – zero, PS – positive small, PM – positive medium, PB – positive big.

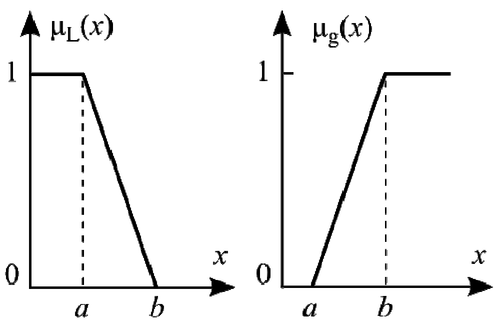

Fig. 10. *L* and *g* fuzzy sets.

$$\mu_L(x,a,b) = \begin{cases} 1 & for \quad x \le a \\ \dfrac{b-x}{b-a} & for \quad a < x \le b \\ 0 & for \quad x > b \end{cases} \tag{4}$$

$$\mu_\gamma(x,a,b) = \begin{cases} 0 & for \quad x \le a \\ \dfrac{x-a}{b-a} & for \quad a < x \le b \\ 1 & for \quad x > b \end{cases} \tag{5}$$

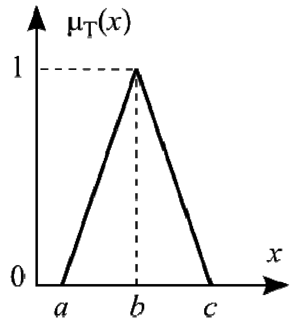

Fig. 11. Triangular type fuzzy set.

$$\mu_\gamma(x,a,b) = \begin{cases} 0 & for \quad x \le a \\ \dfrac{x-a}{b-a} & for \quad a < x \le b \\ \dfrac{c-x}{c-b} & for \quad b < x \le c \\ 0 & for \quad x > c \end{cases} \tag{6}$$

Rule Base consists of 49 rules numbered from i = 1…49, in form of “if-else” conditional expressions. Exemplary rule can be found below

Rn: IF ($e(n) = A_j$) AND ($\delta e(n) = B_k$) THEN ($\Delta u_i = C_i$)

Rules *Rn* bind together both input variables and output variable in cause-effect relations. Table 1 contains all possible control rules. This arrangement is the so called MacVicar-Whelan rule base and is very common for fuzzy logic PI regulators [17, 18]. It was assumed that it is optimal, and it was not a subject of tuning later on

Table 1. Rule base of FLCPI regulator.

e \ δe	NB	NM	NS	Z	PS	PM	PB
NB	NB$_{(0)}$	NB$_{(1)}$	NB$_{(2)}$	NB$_{(3)}$	NM$_{(4)}$	NS$_{(5)}$	Z$_{(6)}$
NM	NB$_{(7)}$	NM$_{(8)}$	NM$_{(9)}$	NM$_{(10)}$	NS$_{(11)}$	Z$_{(12)}$	PS$_{(13)}$
NS	NB$_{(14)}$	NM$_{(15)}$	NS$_{(16)}$	NS$_{(17)}$	Z$_{(18)}$	PS$_{(19)}$	PM$_{(20)}$
Z	NB$_{(21)}$	NM$_{(22)}$	NS$_{(23)}$	Z$_{(24)}$	PS$_{(25)}$	PM$_{(26)}$	PB$_{(27)}$
PS	NM$_{(28)}$	NS$_{(29)}$	Z$_{(30)}$	PS$_{(31)}$	PS$_{(32)}$	PM$_{(33)}$	PB$_{(34)}$
PM	NS$_{(35)}$	Z$_{(36)}$	PS$_{(37)}$	PM$_{(38)}$	PM$_{(39)}$	PM$_{(40)}$	PB$_{(41)}$
PB	Z$_{(42)}$	PS$_{(43)}$	PM$_{(44)}$	PB$_{(45)}$	PB$_{(46)}$	PB$_{(47)}$	PB$_{(48)}$

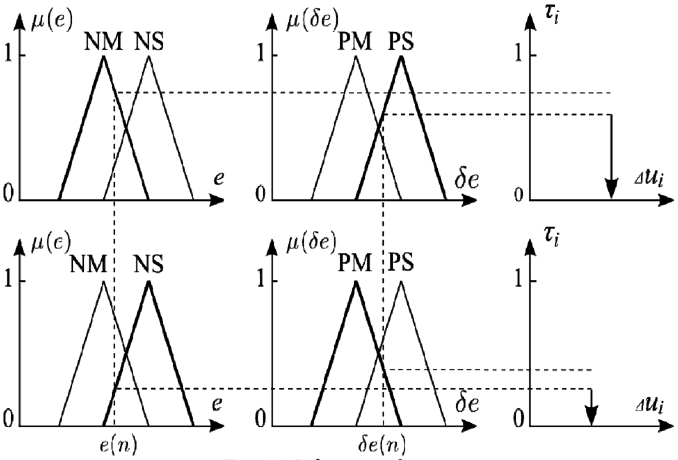

Fig. 12. Inference scheme.

Figure 11 presents the inference scheme of used controller for two exemplary rules. In first step the inputs of error and increase of error are fuzzyfied. Algorithm determines to which fuzzy sets they belong and calculates the value of membership function for each regarded set. Next, the firing levels τ_i for each rule are calculated using the t-norm *min*(.). To receive the crisp output from the regulator, the weighted average defuzzy-fication method expressed below was used:

$$\Delta u(n) = \frac{\sum_{1}^{i} \tau_i \cdot \Delta u_i}{\sum_{1}^{i} \tau_i} \tag{6}$$

where: $\Delta u(n)$ – crisp output, τ_i – firing level of fuzzy rule, Δu_i – output of the fuzzy rule, i – number of the fuzzy rule.

4. Test results

The test was carried out for the setpoint speed of belt pulley at the value of 5, 10 and 15 rpm. Figures 13 to 18 show the best achieved step responses for both regulators. In Table 3 a short comparison of their performance regarding the setpoint time and a sum of absolute values of error over time is provided. Features like execution time of the algorithm and number of tuned variables were compared.

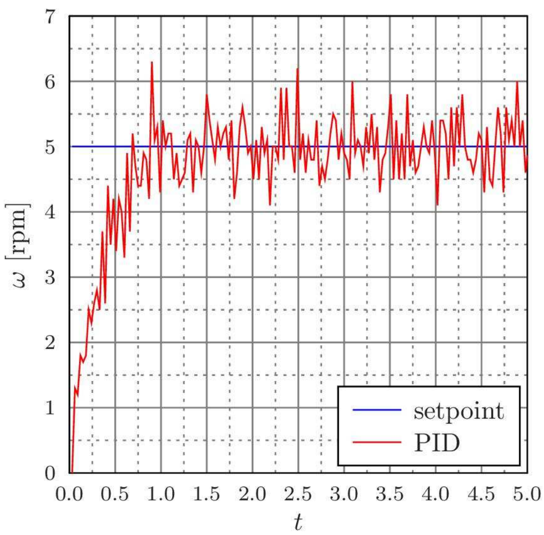

Fig. 13.Time history of the angular velocity controlled by the tuned PID algorithm (K_P=1.2, K_I=2.9, K_D=1.35) and setpoint speed ω_{sp} = 5 rpm.

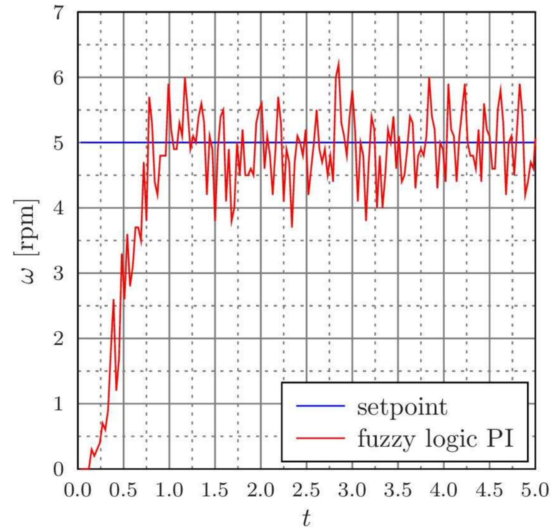

Fig. 14. Time history of the angular velocity controlled by the tuned fuzzy logic PI algorithm and setpoint speed ω_{sp} = 5 rpm.

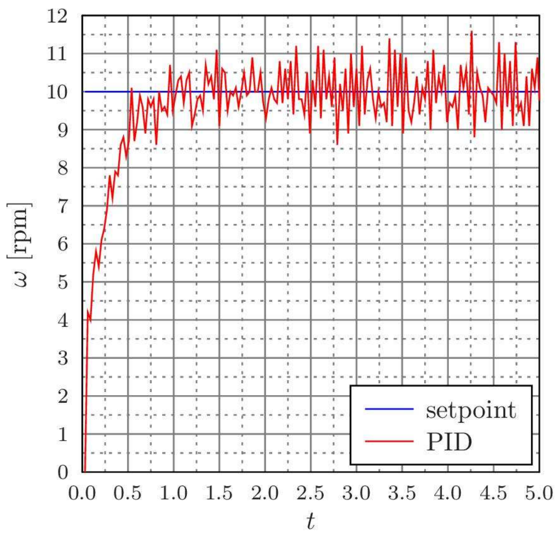

Fig. 15. Time history of the angular velocity controlled by the tuned PID algorithm (K_P = 1.2, K_I = 2.9, K_D = 1.35) and setpoint speed ω_{sp} = 10 rpm.

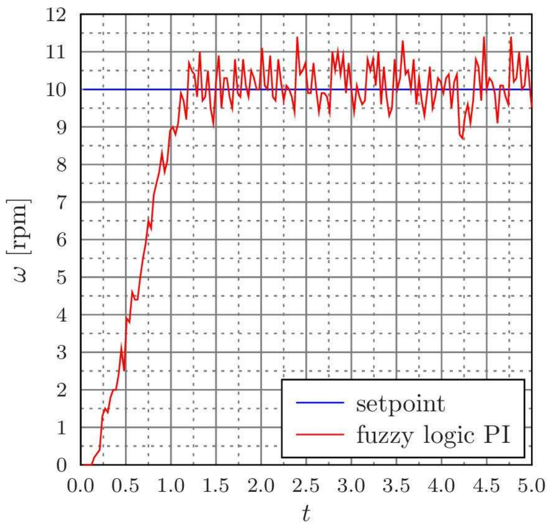

Fig. 16. Time history of the angular velocity controlled by the tuned fuzzy logic PI algorithm and setpoint speed ω_{sp} = 10 rpm.

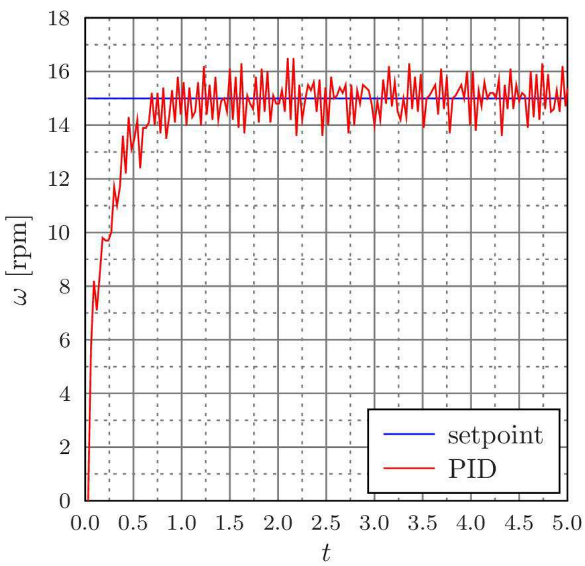

Fig. 17. Time history of the angular velocity controlled by the tuned PID algorithm (K_P=1.2, K_I=2.9, K_D=1.35) and setpoint speed ω_{sp} =15 rpm.

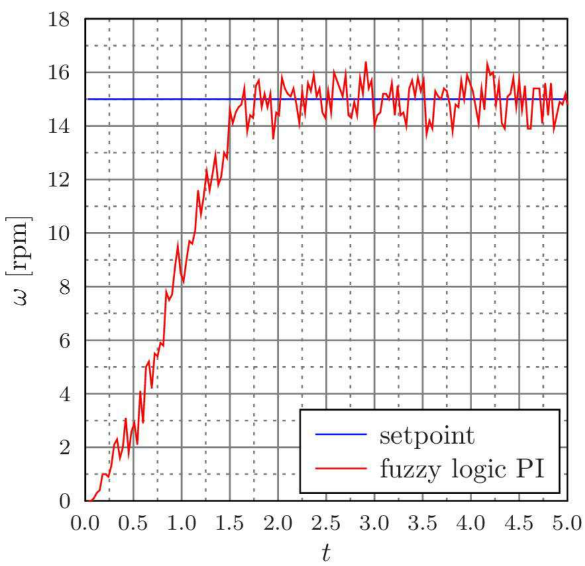

Fig. 18. Time history of the angular velocity controlled by the tuned fuzzy logic PI algorithm and setpoint speed ω_{sp} =15 rpm. Parameters of the FLC PI are given in Table 2.

Table 2. Parameters of used FLC PI regulator.

	NB	NM	NS	Z	PS	PM	PB
e	a=-0.9; b=-0.6	a=-0.9; b=-0.6; c=-0.3	a=-0.6; b=-0.3; c=0.0	a=-0.3; b=0.0; c=0.3	a=0.0; b=0.3; c=0.6	a=0.3; b=0.6; c=0.9	a=0.6; b=0.9
δe	a=-0.6; b=-0.4	a=-0.6; b=-0.4; c=-0.2	a=-0.4; b=-0.2; c=0.0	a=-0.2; b=0.0; c=0.2	a=0.0; b=0.2; c=0.4	a=0.2; b=0.4; c=0.6	a=0.4; b=0.6
u_i	-10	-6	-3	0	3	6	10

Table 3. Performance comparison of considered algorithms.

Property	PID	fuzzy logic PI
Set-point time	0.93 s – 5 rpm 0.96 s – 10 rpm 0.96 s – 15 rpm	0.84 s – 5 rpm 1.12 s – 10 rpm 1.65 s – 15 rpm
Sum of absolute values of error over time	101.9 – 5 rpm 153.3 – 10 rpm 197.2 – 15 rpm	143 – 5 rpm 277.1 – 10 rpm 516.2 – 15 rpm
Lines of code required	105	280
Number of tuned variables	3	45+49 = 94
Execution time of algorithms	73 [μs]	112 [μs]

As can be observed only marginal gain in stability of regulated system was achieved. Both regulators fail to respond properly to chaotic changes in load parameters. Comparison in Table 3 shows that the only parameter in which the fuzzy logic PI regulator exceeds the PID regulator is the time of reaching the first setpoint speed. Due to the fact that the output of the controller is an increase in duty cycle of the PWM signal, a short period at the start of the step response can be observed. It is caused by slow build-up of voltage needed to overcome the forces of static friction. The PID regulator almost instantly provides the necessary value of PWM duty cycle to move the engine's shaft.

An interesting thing can be observed comparing the sum of absolute values of error and the shape of step responses shown in Fig. 14 and 15. Fuzzy logic regulator achieved a lower quality of regulation when the sum of absolute values of error is regarded, yet the shape presented in Fig. 15 appears to be smoother and with less high frequency oscillations than the PID controller. This is due to natural ability of fuzzy logic controllers to handle uncertain information. On the other hand, it seems that the PID controller is interfering in a small way in its own operation. For all of the setpoint speed values (Fig. 13-18) a steady decrease in control quality can be observed for the FLC PI regulator. The time needed to reach the setpoint speed nearly doubles between the first and last trial for the fuzzy logic regulator, and the sum of absolute values of error grows to a value of 360% of the first trial. In case of the classical PID controller the time increases imperceptibly and the sum of absolute values of error grows by 50%.

An idea to decrease the time of setpoint speed would be to introduce an additional fuzzy rule. The regarded fuzzy set would place itself before NB set of error input variable. The rule covering this set would generate a substantial output that would help with gaining speed by the DC motor. Applying a fuzzy PID regulator could also help with the sluggish response of the system, but the exponentially growing number of variables to tune could be problematic. Another idea would be to develop a hybrid controller. For the large values of error of regulation it could act as a regular PID controller, and for lower values near the setpoint the fuzzy logic controller would took the control on the system's stabilization.

5. Conclusions

From an engineering point of view the FLC PI regulator proposes a bigger challenge than any classical PID controller. While in nowadays the computation time and the volume of code is in most of cases insignificant, the amount of the tuned variables proves to be a great downside of the fuzzy logic approach. In Table 3 the amount of tuned variables is given by a sum of two values. Number 45 reflects the amount of parameters describing input and output fuzzy sets, and the 49 is the amount of fuzzy rules in the Rule Base. For the purpose of this work only the first set of variables was tuned, and the Rule Base was assumed to be optimal. Because the variables of the tuned system have clear physical interpretations, the tuning process seems to be very intuitive. Unfortunately, in reality, it demands a broad knowledge about the controlled object as well as many attempts of tuning of control algorithms.

The fuzzy logic controller described in this paper provides lesser quality of control comparing to the classical solution. Despite the interesting concept standing behind it, no substantial improvement in any regarded criteria was achieved. Additionally, the very complex tuning process extends the time in which promising results were obtained. To reduce the difficulty of tuning, the fuzzy logic approach requires its own learning algorithm which would pick the right parameters and adopt them actively to the experiment.

References

1. L. A. Zadeh, “Knowledge Representation in Fuzzy Logic”, *IEEE transactions on knowledge and data engineering*, vol. 1, no. 1, 1989, pp. 89-101.
2. L. A. Zadeh, “Is there a need for fuzzy logic?”, *Information Sciences*, 178, 2008, pp. 2751-2779.
3. Y. Bai, D. Wang, “Fundamentals of Fuzzy Logic Control – Fuzzy Sets, Fuzzy Rules and Defuzzifications”, [in:] *Advanced Fuzzy Logic Technologies in Industrial Applications*, Eds. Y. Bai, H. Zhuang, D. Wang, 2006, London, Springer London, pp. 17-36.
4. R. Jager, "Fuzzy logic in control", Ph.D thesis, T.U. Delft, 1995
5. P. Neto, N. Mendes, J.N. Pires, J. Norberto, A.P. Moreira, “CAD-Based Robot Programming: the role of Fuzzy-PI Force Control in Unstructured Environments” [in:] *IEEE Conference on Automation Science and Engineering*, Toronto, 2010, pp. 362 - 367.
6. J. Velagic, A. Galijasevic, “Design of Fuzzy Logic Control of Permanent Magnet DC Motor under Real Constraints and Disturbances”, *IEEE Multi-conference on Systems and Control*, Saint Petersburg, 2009, pp. 461 - 466.
7. K.C. Chan, X. Chu, “Design of a Fuzzy PI Controller to Guarantee Proportional Delay Differentiation on Web Servers”, *Algorithmic Aspects in Information and Management*, Portland, 2009, pp. 389 - 398.
8. J.T. Teeter, M. Chow, J.J. Brickley, “A Novel Fuzzy Friction Compensation Approach to Improve the Performance of a DC Motor Control System”, *IEEE Transactions on Industrial Electronics*, Vol. 43, No. 1, February 1996, pp. 113-120.
9. M. Bašić, D. Vukadinović, M. Polić, “Fuzzy Logic vs. Classical PI Voltage Controller for a Self-Excited Induction Generator”, *Mathematical applications in science and mechanics*, Eds. Natasa Trisovic, Deolinda Rasteiro, Dubrovnik,WSEAS Press, 2013, pp. 189-194.
10. J. -S.R. Jang, “Fuzzy modeling using generalized neural networks and Kalman filter algorithm”, *AAAI'91: Proceedings of the ninth National conference on Artificial intelligence*, vol. 2, Anaheim, 1991, pp. 762-767.
11. Jee S., Korem Y., “Adaptive fuzzy logic controller for feed drives of a CNC machine tool”, Mechatronics, 14, Elsevier, 2004, pp. 299–326.
12. M. Arrofiq, N. Saad, "A PLC-based Self-tuning PI-Fuzzy Controller for Linear and Non-linear Drives Control", IEEE International Conference on Power and Energy, Johor Bahru, 2008, pp.701-706.
13. W. Kunikowski, J. Awrejcewicz, P. Olejnik, “Efficiency of a PLC-based PI controller in stabilization of a rotational motion affected by the chaotic disturbances”, *Dynamical Systems –Applications*, Eds. J. Awrejcewicz, M. Kaźmierczak, P. Olejnik, J. Mrozowski, Publishing House of Lodz University of Technology, 2013, pp 173-184.
14. J. Awrejcewicz, P. Olejnik, “Friction pair modeling by 2-dof system: numerical and experimental investigations”, *International Journal of Bifurcation and Chaos*, World Scientific, 15(6), 2005, pp 1931-1944.
15. J. Awrejcewicz, P. Olejnik,“Occurrence of stick-slip phenomenon”, *Journal of Theoretical and Applied Mechanics,* 45(1), 2007, pp 33-40.
16. R. Petrella, M. Tursini, L. Peretti, M. Zigliotto, Speed Measurement Algorithms for Low-Resolution Incremental Encoder Equipped Drives: a Comparative Analysis, ACEMP '07. International Aegean Conference on Electrical Machines and Power Electronics, Bodrum, 2007, pp. 780 - 787.
17. F. Cheong, R. Lai, ”Simplifying the automatic design of a fuzzy logic controller using evolutionary programming”, Soft Computing, Volume 11, Issue 9, July 2007 pp 839-846.
18. P.J. MacVicar-Whelan, “Fuzzy sets for man-machine interactions”, International Journal of Man-Machine Studies, vol. 8, no. 6, June 1977, pp. 687–697.