RECENT J. (2018), 58:070-076
https://doi.org/10.31926/RECENT.2019.58.070

Fuzzy vs PID Controller: Design Management

Catrina CHIVU
Transilvania University of Brasov, Romania, catrina.c@unitbv.ro
Cătălin-Iulian CHIVU
Transilvania University of Brasov, Romania, catalin.c@unitbv.ro

Abstract
Fuzzy logic was, first time presented by L.A. Zadeh in 1972 as an alternative method of classical theory control, by giving the opportunity to use a non-analytic method. During time, fuzzy logic was extensively applied in automotive, aerospace, business, electronic and even defence field, considering that is an easier way to obtain high performance. Automatic experts are constantly researching and proposing innovative and effective fuzzy control systems. The problem appears not in simulating the problem but integrating controller in an electric circuit. Therein lies the problem of the profitability of the fuzzy controller comparative to a classic PID control. Besides the time and knowledge requirements are quite different.

Keywords
robotics, gripper, fuzzy controller, PID

1. Problem Description and Application Fields

Fuzzy and PID controllers have a variety of applications starting with automotive field, through medicine (especially fuzzy combined with neural and genetic algorithms) [1]. The literature in fuzzy control has been growing rapidly in recent years, making it difficult to present a comprehensive survey of the wide variety of applications that have been made [2].Thus, according to literature there are fuzzy controllers used in topology, decision processes, medicine, mathematics, numeric methods, etc. [1]. As a remark relative to the fields, are those were phenomena are not well-known or are not well-determined as time variation.

On the other hand, the PID controller was first mentioned around 1911 by Elmer Sperry, being, until this moment, the most used and applied type of controller. The main advantage of PID controller is that requires minimal background and user's effort and – albeit limited – may shape the system's both transient and asymptotic performance.

Thus, the problem arises in using the modern, "what's trendy", or is the classic just as useful? Present paper tries to find out the answer to this question. In fact, the real problem is related to design time and knowledge requirements. Thus, designing a controller should be a short-time process and should requires as little knowledge as possible. This is why the authors of this paper have tried to compare the two controllers in terms of efficiency.

Present paper has stopped on an example of a mechanical gripper controlled by both fuzzy and PID controllers. The idea was that a mechanical gripper may be designed using linkage mechanism that are used also for industrial robots or manipulators and thus, the applicability goes beyond its strict use. Grippers are very complex mechanical devices, and their design has to satisfy a large number of constrains: gripping force, movement trajectory of the fingers, shape of the gripped objects and so on [3]. The fingers design begins considering the numbers of joints, the mechanisms chosen to model the finger and the lengths of each element.

To be able to compare the two controllers it is used Matlab simulation program and its toolboxes. The mechanical structure is a robotic mechanical gripper, with two fingers, used by a pick-and-place manipulator. Gripper will be considered actuated by a PWM control DC motor. To compare the results the interface is developed also in Matlab, and fuzzy and PID controllers are those given by the software toolboxes. Using Matlab as simulation software, the part of modelling of the mechanical component is

easier to be done, because of the Simscape toolbox that has components as: linkage, joint, primary mechanisms. Mechanism may be also designed in Creo Parametric3D Modelling Software, saved as *.csv file and imported in Matlab (at least in versions after 2016b).

2. Design Controllers

2.1. Fuzzy controller

Fuzzy logic is the logic on which control is based to human thinking and natural language. In fact, fuzzy logic is not a succession binary conditions, but a set of linguist control rules, that can vary between 0 to 1 along a well-known rule. This method focuses on what the system should do rather than trying to understand how it works. One can concentrate on solving the problem rather than trying to model the system mathematically, if that is even possible. This almost invariably leads to quicker, cheaper solutions. Once understood, this technology is not difficult to apply and the results are usually quite surprising and pleasing [4].

Fuzzy theory celebrated its 50th anniversary in 2015. Thus, the main benefits of fuzzy is the concept: sets rather than only degrees of truth. In other words, while Boolean algebras are underlying both propositional logic and naive set theory, the set point of view may be found richer in terms of mathematical modelling, and the same thing takes place when moving from many-valued logics to fuzzy sets [1].

To develop a fuzzy logic controller there should be followed the steps:

- identification of variables: in this case input variables were the movements and forces from the input of mechanism, PWM data, actuator parameters; outputs are movement of the fingertip and gripping forces;
- fuzzy subset configuration: information is divided in fuzzy subsets;
- obtaining membership function: associate a function to each fuzzy subset;
- fuzzification;
- combining fuzzy outputs: locate the fuzzy output and merge them;
- defuzzification.

In the following is designed the fuzzy controller, which, in the case of gripper is a Takagi-Sugeno controller. The designed fuzzy controller has as inputs the angular position error and differential of angular position error and as output the PWM duty factor. The two inputs "e" and "ė" are determined, at any moment, using the equations [5]:

$$\begin{aligned} e(k) &= n^*(k) - n(k) \\ \dot{e}(k) &= e(k) - e(k-1) \end{aligned} \tag{1}$$

where $n^*(k)$ is the reference angular position at that moment and $n(k)$ is the determined actual position.

The two linguistic variables, corresponding to the inputs, have five linguistic terms: NM – negative medium, NS – negative small, ZE – zero, PS – positive small, PM – positive medium. The membership functions are triangular one and their definition are shown in Figure 1. The PR1X, PR2X, PR3X, CX and PR1V, PR2V, PR3V, CV parameters define the definition interval of the two membership functions (PR1X – a1, PR2X – a2, PR3X – a3, PR1V – a1, PR2V – a2, PR3V – a3) and, respectively, the maximums of PM and NM intervals (CX, CV). To these parameters where attributed an estimative set of values and, after controller implementation those values where modified in the simulation process.

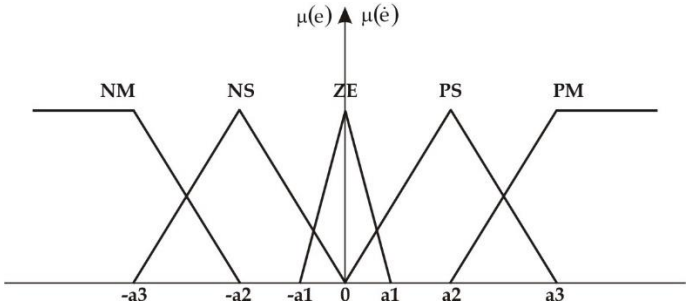


Fig. 1. Membership functions of the fuzzy controller

The constants of the fuzzy controller are symbolized by S_{x1}, S_{x2}, S_{x3} and S_{v1}, S_{v2}, S_{v3}. Using these notations where defined the rules R1, …, R25 that were used to determine the numeric values of the fuzzy controller outputs (Table 1).

Table 1. Fuzzy controller outputs

		ė				
		NM	NS	ZE	PS	PM
e	*NM*	$-S_{x1}\cdot x + S_{v1}\cdot v$	$-S_{x1}\cdot x + S_{v2}\cdot v$	$-S_{x1}\cdot x + S_{v3}\cdot v$	$-S_{x1}\cdot x + S_{v2}\cdot v$	$-S_{x1}\cdot x + S_{v1}\cdot v$
	NS	$-S_{x2}\cdot x + S_{v1}\cdot v$	$-S_{x2}\cdot x + S_{v2}\cdot v$	$-S_{x2}\cdot x + S_{v3}\cdot v$	$-S_{x2}\cdot x + S_{v2}\cdot v$	$-S_{x2}\cdot x + S_{v1}\cdot v$
	ZE	$-S_{x3}\cdot x + S_{v1}\cdot v$	$-S_{x3}\cdot x + S_{v2}\cdot v$	$-S_{x3}\cdot x + S_{v3}\cdot v$	$-S_{x3}\cdot x + S_{v2}\cdot v$	$-S_{x3}\cdot x + S_{v1}\cdot v$
	PS	$-S_{x2}\cdot x + S_{v1}\cdot v$	$-S_{x2}\cdot x + S_{v2}\cdot v$	$-S_{x2}\cdot x + S_{v3}\cdot v$	$-S_{x2}\cdot x + S_{v2}\cdot v$	$-S_{x2}\cdot x + S_{v1}\cdot v$
	PM	$-S_{x1}\cdot x + S_{v1}\cdot v$	$-S_{x1}\cdot x + S_{v2}\cdot v$	$-S_{x1}\cdot x + S_{v3}\cdot v$	$-S_{x1}\cdot x + S_{v2}\cdot v$	$-S_{x1}\cdot x + S_{v1}\cdot v$

From Table 1 it can be concluded that the fuzzy set of rules can be decomposed, based on the two input variables (angular position error and angular speed error). Thus, for angular position error, *x*, the rules are presented in Table 2 and the rules for angular speed error, *v*, in Table 3.

Table 2. Fuzzy rules for position error

NM	$S_{x1}\cdot x$
NS	$S_{x2}\cdot x$
ZE	$S_{x3}\cdot x$
PS	$S_{x2}\cdot x$
PM	$S_{x1}\cdot x$

Table 3. Fuzzy rules for angular speed error

NM	$S_{v1}\cdot v$
NS	$S_{v2}\cdot v$
ZE	$S_{v3}\cdot v$
PS	$S_{v2}\cdot v$
PM	$S_{v1}\cdot v$

Thus, the numeric value of the output that should be computed by the controller is: "output" = "angular speed correction" – "angular position correction". Based on Figure 1, there can be written the computing mode and the value of the angular position error (Table 4).

Table 4. Angular position error: computing mode and value

Interval	*Computing mode*	*Value*
NM	NM	$S_{x1}\cdot x = C_{1x}\cdot x$
NM∩NS	NM+NS	$S_{x1}\cdot x + S_{x2}\cdot x = C_{1x}\cdot x$
NS	NS	$S_{x2}\cdot x = C_{2x}\cdot x$
NS∩ZE	NS+ZE	$S_{x2}\cdot x + S_{x3}\cdot x = C_{3x}\cdot x$
ZE	ZE	$S_{x3}\cdot x = C_{4x}\cdot x$
ZE∩PS	ZE+PS	$S_{x2}\cdot x + S_{x3}\cdot x = C_{3x}\cdot x$
PS	PS	$S_{x2}\cdot x = C_{2x}\cdot x$
PS∩PM	PS+PM	$S_{x1}\cdot x + S_{x2}\cdot x = C_{1x}\cdot x$
PM	PM	$S_{x1}\cdot x = C_{1x}\cdot x$

From Fig. 1 result the angular correction (AngCorr):

If $x < -PR_{3x}$ then *angCorr* is $C_x\cdot x$

If $-PR_{3x} \le x < -PR_{2x}$ then *angCorr* is $C_{1x}\cdot x$

If $-PR_{2x} \le x < -PR_{1x}$ then *angCorr* is $C_{2x}\cdot x$

If $-PR_{1x} \le x < 0$ then *angCorr* is $C_{3x}\cdot x$

If $x = 0$ then *angCorr* is $C_{4x}\cdot x$

If $0 < x \leq PR_{1x}$ then $angCorr$ is $C_{3x} \cdot x$
If $PR_{1x} < x \leq PR_{2x}$ then $angCorr$ is $C_{2x} \cdot x$
If $PR_{2x} < x \leq PR_{3x}$ then $angCorr$ is $C_{1x} \cdot x$
If $PR_{3x} < x$ then $angCorr$ is $C_x \cdot x$

The above rules can be simplified by observing that:

- when angular error is x = 0, the constant $C_{4x} \cdot x$ may take any finite value. Thus, this rule can be included in one or both adjacent rules;
- for the extreme rules ($x < -PR_{3x}$ or $PR_{3x} < x$) the resulted values for correction are very big in modulus, relatively to numeric limits for the variation interval of PWM duty factor (the output of fuzzy controller). These values will be truncated, thus the angular correction will be considered $C_x \cdot x$;
- the rules are symmetrical relative to x = 0.

Considering these observations the rules become:

If $0 \leq |x| \leq PR_{1x}$ then $angCorr$ is $C_{3x} \cdot x$
If $PR_{1x} < |x| \leq PR_{2x}$ then $angCorr$ is $C_{2x} \cdot x$
If $PR_{2x} < |x| \leq PR_{3x}$ then $angCorr$ is $C_{1x} \cdot x$
If $PR_{3x} < |x|$ then $angCorr$ is $C_x \cdot sgn(x)$

The same analysis and simplification can be done for the angular speed error. Thus, the final rules are:

If $0 \leq |v| \leq PR_{1v}$ then $angCorr$ is $C_{3v} \cdot v$
If $PR_{1v} < |v| \leq PR_{2v}$ then $angCorr$ is $C_{2v} \cdot v$
If $PR_{2v} < |v| \leq PR_{3v}$ then $angCorr$ is $C_{1v} \cdot v$
If $PR_{3v} < |v|$ then $angCorr$ is $C_v \cdot sgn(v)$

Where PR_{1x}, PR_{2x}, PR_{3x} are the domains of the membership functions of the angular position error and PR_{1v}, PR_{2v}, PR_{3v} represents the domains of the membership functions of the angular speed error (see a_1, a_2 and a_3 in Fig. 1). These values can be modified after the experiments or simulations.

The values for C_{1x}, C_{2x}, C_{3x} and C_{1v}, C_{2v}, C_{3v} are constants and were chosen as powers of 2 (to minimize the execution time of the microcontroller program). C_x and C_v are the outputs of extreme rules ($PR_{3x} < x$ and $PR_{3v} < v$) and can be modified based on experiments and simulations.

The output of the fuzzy controller is the difference between the angular speed correction and angular position correction. This output is then linearly corrected relative to instantaneous angular acceleration. The result is truncated to the limit of PWM duty factor domain (the output of the controller). Thus, the maximum motor torque is controlled limited in both rotation senses.

These two sets of rules will be easier to be implemented in the microcontroller's program, which has as critical resource the execution time of a minimal module (module that cannot have interrupt) between two successive samplings of the signals given by the incremental transducer.

2.2. PID controller

Proportional Integral Derivative (PID) control is one of the first control strategies use in engineering. At the beginning it was implementing in analog electronics and pneumatic devices. In the present, is still the most used classical controller. PID is characterized by the equation (2):

$$u(t) = K_P \cdot \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \cdot \frac{de(t)}{dt} \right] \quad (2)$$

where $u(t)$ is the input signal; $e(t)$ – error signal, computed as difference between reference signal and output; K_P – proportional coefficient; T_i – integral coefficient; T_d – derivative coefficient.

In practice, there are three types of PID controller: zero, first and second order, based on the delay of the input (in fact, there is a proportional input, a first order derivate input or second order derivate input).

The quality of a PID controller depends on tuning [6]. In Matlab there are specific functions for each tuning methods. Thus, there is Ziegler-Nichols formula (Matlab function - *ziegler()*), Chien–Hrones–

Reswick PID Tuning Algorithm (Matlab function - *chrPID()*), Cohen–Coon Tuning Algorithm (Matlab function - *cohenpid()*), Wang–Juang–Chan Tuning Formula (Matlab function - *wjcpid()*), Disturbance rejection (Matlab function - *optpid()*). Tuning the controller is a very time-consumption process and a knowledge-based process. Besides these methods, depending on designer background, tuning may be done also based on transfer function, or Nyquist diagram, discrete response diagram.

2.3. Controllers in Matlab

A Matlab App program was designed to be able to introduce input constant data and to access the two controller, Figure 2.

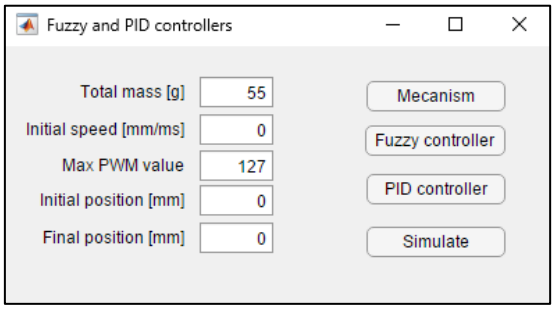
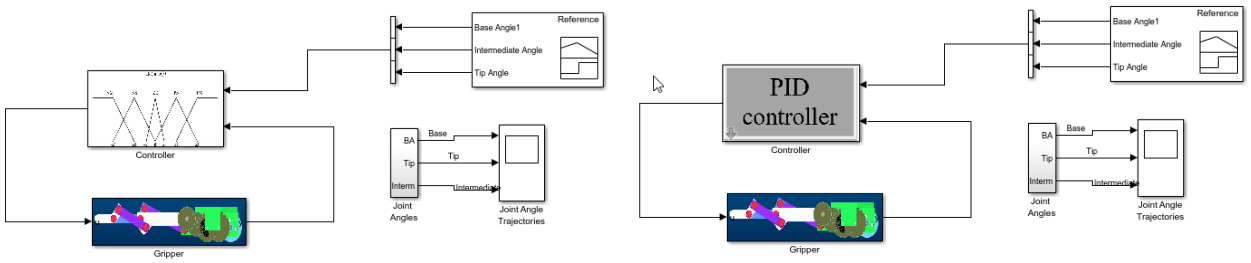


Fig. 2. Matlab App: fuzzy and PID controller

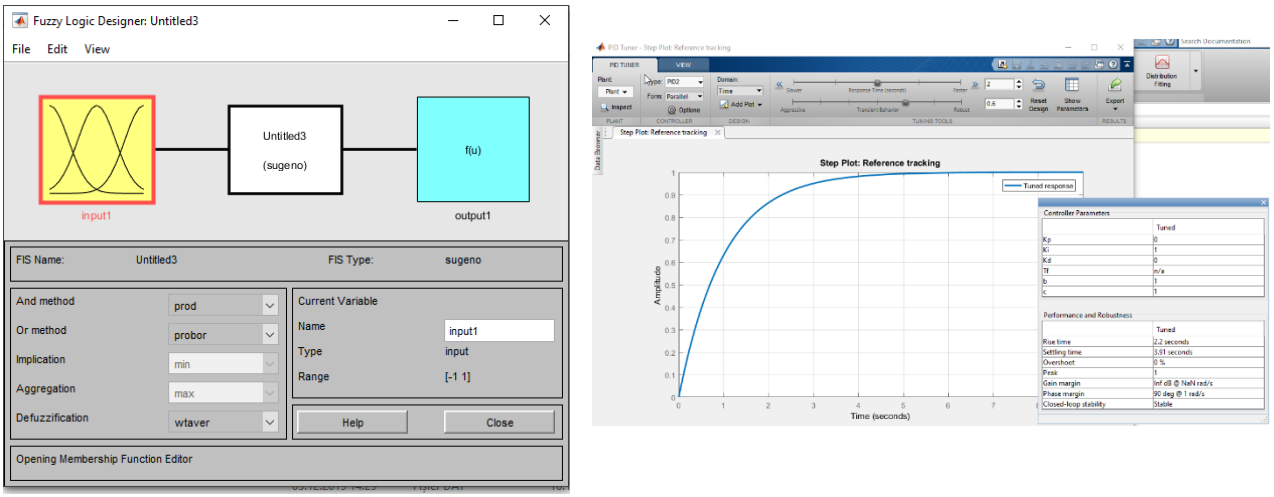
The gripper was design in Creo and imported in Matlab (Figure 3).



a) Fuzzy controller b) PID controller

Fig. 3. Simulation diagrams in Simscape

To be able to correctly compare the two controllers for both of them were used Matlab tools. Thus for fuzzy controller the rules and functions were defined using Fuzzy module (Figure 4), implemented, as it is shown in Figure 3 in a Simscape model.



a) Fuzzy controller b) PID controller

Fig. 4. Matlab fuzzy and PID controllers interface

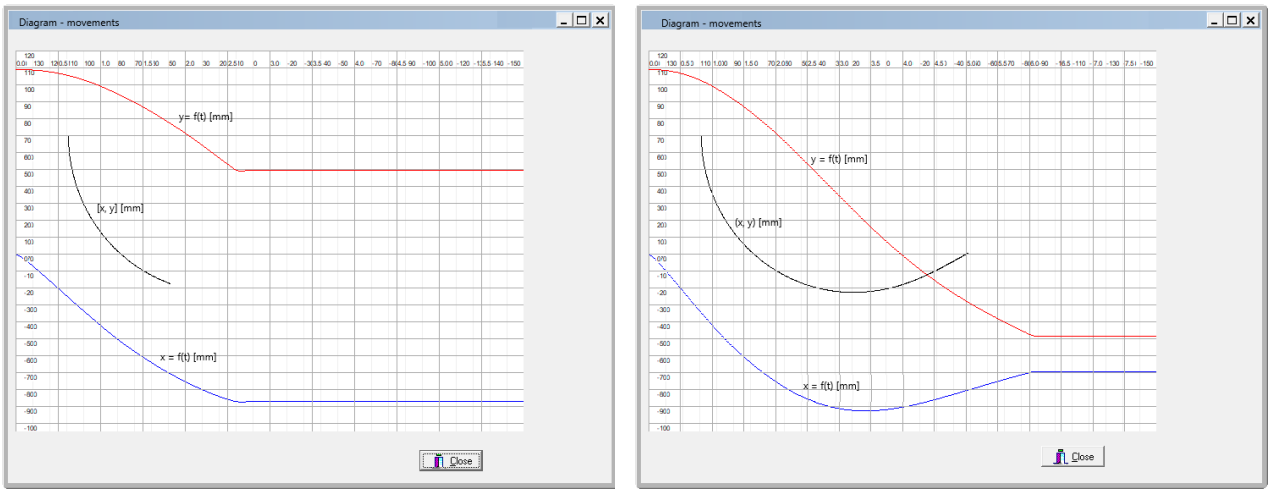
3. Benchmarking

To benchmark the design of the two controllers it is mandatory to do a response analysis, a time analysis and required knowledge analysis. In the following there are presented the methods applied and results.

Response analysis was done considering:

- the mobile mass of each finger: 50 [g];
- adjacent mass of the systems: 30 [g];
- actuator maximum torque: 4.4 [mNm];
- initial value of PWM duty factor: 50 [%];
- maximum deviation of PWM duty factor: 50 [%];
- static friction: 0.05 [mNm];
- coefficient of internal voltage: 0.0244;
- transducer step: 1.38 [mm].

Simulations were done in the case of cylinder and total-close grasping. Trajectories for the fingertip are presented in Figure 5.



a) Cylinder grasping b) total-close grasping

Fig. 5. Desired trajectories of the fingertip

Dynamic parameters of simulations after tuning both controllers are presented in Table 5.

Table 5. Response dynamic parameters

Controllers / Dynamic parameters	Fuzzy controller	PID controller
Settling time [s]	4.32	4.35
Overshoot [%]	6	2
Rise time [s]	1.67	2.01
Delay time [s]	0.84	1

Analysing the results given in Table 5, PID controller gives better results, even if time response are higher, has a better damping. After auto-tuning of PID controller the results were even better. To be able to obtain a better damping for fuzzy controller the rules definition should be refined, but the results depends on the experience of the designer. However, the systems responses are similar and the decision regarding which controller is better is on designer hand.

To be able to compare the two from the other points of view, there is done a multicriterial analysis, presented in Table 6. Marks presented in Table 6 are average of those given by specialist who use both controllers (specialist should give a mark between 1 and 10).

Table 6. Multicriterial analysis

	Importance coefficient	Fuzzy controller	PID controller
Dimension of data sets	0.15	4.6	8.83
Work with imprecisely data sets	0.2	9.3	6.7
Robustness	0.1	8.76	6.4
Human thinking	0.05	9.3	3.4
Human expertise	0.15	3.2	8.7
Design time	0.15	4.67	7.8
Continuous updating tuning	0.15	3.78	9.5
Feasibility	0.05	7.65	8.76
	Total	6.021	7.8125

4. Conclusions

Multicriterial analysis gives, apparently a very clear result: PID controller is better than a fuzzy controller. This results is very true if we considered a very well-known system/ plant (Matlab term), relatively steady-system. It is also recommended if the system is very sensible to overshoot (input or perturbation), but it is acceptable a higher steady-time. In fact, if we have a classical mechanical system (as a linkage gripper or mechanism) a PID controller is easier and cheaper to develop and implement.

On the other hand, fuzzy controller is more appropriate if acquired date are not quite precise, process is an intuitive one, or based on human experience (as diagnostic control), it is required a feasible and robust control, even if it is necessary a continuous updating. At least for the last disadvantage the solution to reduce it is to develop a neural-fuzzy controller that auto-updates.

References

1. Dubois D., Prade H. (2015): *Fuzzy logic in its 50th year*. Springer Publisher, ISBN 978-3-319-31091-6
2. Ross T. (2016): *Fuzzy logic with engineering applications*. Wiley Publisher, ISBN 978-1119235866
3. O'Toole M., Bouazza-Marouf K., Kerr D., Vloeberghs M. (2013): *Robust contact force controller for slip prevention in a robotic gripper*. Proc. of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, ISSN 0959-6518, Vol. 224, is. 3, p. 275-288, http://dx.doi.org/10.1243/09596518JSCE854
4. Aissaoui A.G., Tahour A. (2012): *Application of fuzzy logic in control of electrical machines*. In: Dadios E. (Ed.) *Fuzzy Logic. Controls, Concepts, Theories and Applications*. IntechOpen, ISBN 978-953-51-0396-7, p. 107-128, DOI: 10.5772/35770
5. Ogata K. (2015): *Modern control engineering*. Pearson India Publisher, ISBN 978-9332550162
6. Xue D., Chen Y.Q, Atherton D. (2009): *Linear feedback control: analysis and design with MATLAB*. Society for Industrial and Applied Mathematics Publisher, ISBN 978-0898716382