EURASIAN JOURNAL OF MATHEMATICAL AND COMPUTER APPLICATIONS
ISSN 2306–6172
Volume 7, Issue 1 (2019) 65 – 78

METHODS OF REDUCING THE COMPUTATIONAL COMPLEXITY OF FUZZY INFERENCE ALGORITHMS FOR IMPLEMENTATION ON A MICROCONTROLLER WITH LIMITED COMPUTATIONAL RESOURCES

Zhartybayeva M.G., Tatur M.M.,Shaverdo M.M., Iskakov K.T.

Abstract In our work we suggest the most effective ways to reduce the complexity of computing fuzzy inference algorithms, which do not lead to a valuable decrease in the control quality. Our suggestions will be analytically formulated, illustrated with examples and results of model experiments.

Key words: fuzzy inference algorithms, fuzzy logic, control system

AMS Mathematics Subject Classification: 03B52, 94D05

DOI: 10.32523/2306-6172-2019-7-1-65-78

Introduction

Along with Proportional-integral-differentiating (PID) regulators, fuzzy controllers are increasingly being used for adaptive non-linear control of various objects. These controllers are based on the device of fuzzy inference, which allows representing and using the expert's knowledge to adjust the parameters of the control system [1]. There is a number of technical and economic requirements for designing fuzzy controllers, the main of which are [2,3]:

- program quality management support [4-8];
- real-time management support [9,10];
- cost minimization of the hardware platform (microcontrollers).

The first requirement can be characterized by such properties, as the smoothness of the change in the controlled parameter, resistance to noise and fluctuations, adaptability, learnability, etc., with unconditional fulfillment of the task. The quality of control depends on the applied method (algorithm) and its successful configuration [1, 4].

The second requirement means that for the predefined object control (actuator) it is necessary to execute the algorithm to control the set number of times per unit time [10, 15].

The third requirement is related to the hardware implementation of control algorithms. The choice of the required type of microcontroller is determined by such

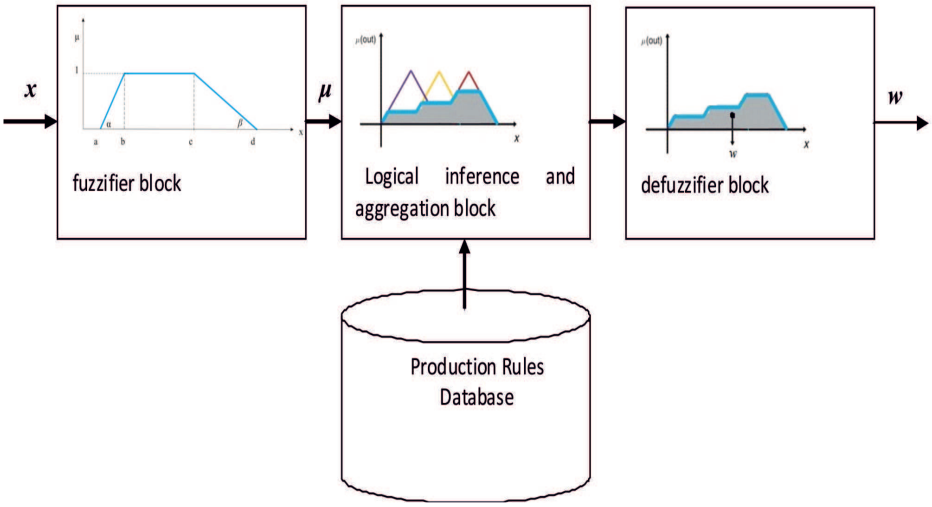


Figure 1: General diagram of fuzzy inference.

characteristics as architecture, bit depth, the amount of internal memory, the presence of built-in multiplication units, etc. As a consequence, the selected element base determines the performance and cost of the fuzzy controller in general [11].

It is known that the implemented control algorithm is characterized by computational complexity and has objective limitations on the execution time for the fixed hardware platform. In other words, making a complex calculation in real time on simple low-power microcontrollers is an immediate problem. In our work we suggest the most effective ways to reduce the complexity of computing fuzzy inference algorithms, which do not lead to a valuable decrease in the control quality. Our suggestions will be analytically formulated, illustrated with examples and results of model experiments.

1 Problem statement

The basis of the fuzzy controller is a well-known conceptual diagram [2]. The measured initial data from sensors monitoring the control process are translated in the meaning of the linguistic variables in the block of the fuzzifier. Further, the procedure of fuzzy inference on the set of production rules that make up the knowledge base of the control system is realized. As a result, the output linguistic variables, which are combined (aggregated) and entered the defuzzifier block, are activated. The control actions on the actuating mechanisms are formed at the output of the defuzzifier (Figure 1)

Performing calculations in each block of the fuzzy controller requires certain computational resources and time to implement the executable algorithms. The total amount of calculations V_{total} in each cycle will include the amount of calculation at each stage, i.e.

$$V_{total} = V_f + V_i + V_a + V_{df} \tag{1}$$

where V_f, V_i,V_a, V_{df} - volumes of calculations at the stages of fuzzifier, inference, aggregation and defuzzifier, accordingly.

Obviously, if the computational complexity of the algorithms is large (the amount of computation is large), then, accordingly, a large calculation time is required. In order to reduce the time with unchanged algorithms it is necessary to increase the processing power of microcontrollers. Therefore, the actual problem is the development of ways to implement fuzzy inference algorithms with minimizing computational complexity (or reducing the amount of computation).

In this article, by the computational complexity of the algorithm we mean the number of basic operations through which the rest of the more complex operations are expressed. For example, if you use the addition operation with the amount of calculations v (and the equivalent operations of subtraction, comparison, min/max, logical bitwise operations, shifts) as the base, the complexity of the multiplication operation for typical Arithmetic Logic devices (ALD) will be mv, where m is the microprocessor's bit depth. Usually the complexity of division operations is estimated as twice the complexity of the multiplication operations - $2mv$, the complexity of exponentiation operations, the calculation of the square root, etc. have their increasing coefficients. The accuracy of establishing these coefficients is rather arbitrary and is not discussed in the present paper, although the direction of the increase in complexity in these operations is obvious.

Note. In a number of applied problems, the computational complexity of the algorithm is also associated with the requirement for the amount of internal memory of the processor. It is clear that using a fast algorithm will not lead to the expected results if its operation requires more RAM (Random Access Memory) than the controller has. However, in this paper we put a restriction on the field of research and focus only on reducing the number of basic operations.

2 Estimation of the amount of computation at the stages of fuzzy inference

The most obvious and well-known way of reducing computational complexity in the algorithm of fuzzy inference is the use of the membership functions of triangular and trapezoidal forms instead of bell-shaped (Gaussian). To calculate the values of the trapezoidal membership function μ, the following formulas are most often encountered in the literature [12-14]:

$$\mu = \begin{cases} 0, a > x, x > d \\ \frac{d-x}{d-c}, c \leq x \leq d \\ 1, b < x < c \\ \frac{x-a}{b-a}, a \leq x \leq b \end{cases} \tag{2}$$

where x ı input value of the measured parameter, a, b, c, d ı parameters of the membership function according to figure 2.

At the stage of fuzzification an extreme case is possible for each input variable k, when it is necessary to calculate the values of two membership functions, figure 3.

Then, according to expressions (2), the amount of calculations in the phase of fuzzification will consist of four subtraction operations and two division operations for each input variable, i.e.

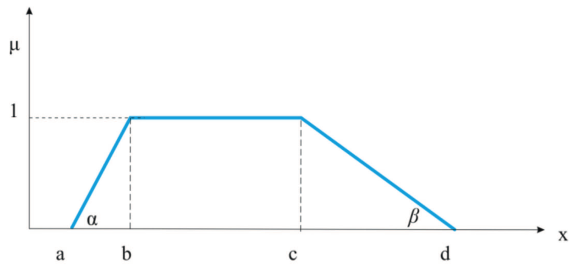


Figure 2: Parameters of the trapezoidal membership function.

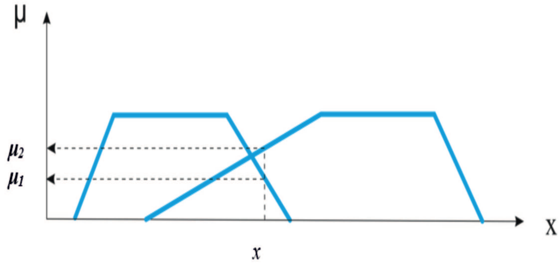


Figure 3: Extreme case, in which the values of two membership functions are calculated.

$$V_f = k(2 * 2v + 2 * 2mv) \tag{3}$$

The stage of inference implies taking the operation of a minimum over all input variables and calculating all rules with truncation (division) by the coefficient μ, then

$$V_i = k * 2mv \tag{4}$$

The aggregation stage includes a join operation, the complexity of which can be taken as v

$$V_a = v \tag{5}$$

The defuzzification stage consists in calculating the value of the output variable w by one of the known criteria (center of gravity, median, etc.). Thus, for a simplified method of defuzzification [15]

$$w = \frac{\sum_i^n \mu_i C_i}{\sum_i^n \mu_i} \tag{6}$$

where each activated rule from the base n corresponds to the constant C_i. Then, the calculation volume includes two cyclic sums and a division operation, i.e.

$$V_{df} = 2nv + 4mv \tag{7}$$

The computational complexity of the defuzzification Vdf for the aggregated complex figure with the criterion of the center of gravity for the classical realization of the Mamdani algorithm [16] is much higher. Therefore, for expert evaluations we will be guided by a simplified way of defuzzification.

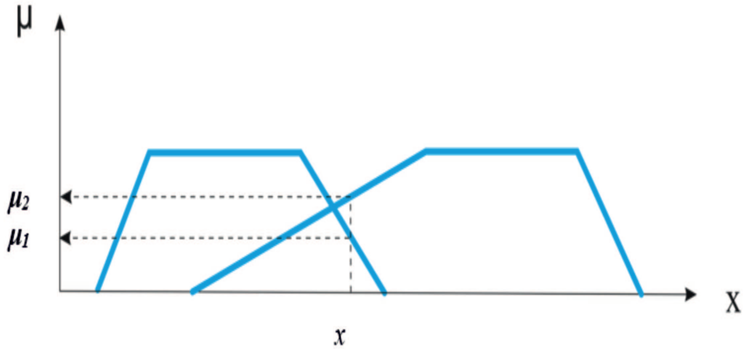


Figure 4: Extreme case, in which the value of one membership function μ_1 is calculated, and the second is defined as: $\mu_2 = 1 - \mu_1$

As can be easily seen from the above analytical estimates, the most complex (capacious) computationally are the stages of fuzzification and defuzzification. So, we further focus on ways to reduce the amount of computation in these components of the time cycle of the control algorithm.

3 Methods of reducing the amount of calculations

In addition to the above, we suggest ways of further reducing the computational complexity of the fuzzy inference algorithm by simplifying the calculations at a lower algorithmic level. At the same time, we do not bring up the problem of improving performance by optimizing the program code and the hardware component.

3.1 Methods of reducing the amount of calculation of the fuzzification stage Vf

If we calculate the coefficients of the membership function (Figure 2) $tg\alpha = \frac{1}{b-a}, tg\beta = \frac{1}{d-c}$ during the fuzzy controller setup (training), and then use it as constants, the calculated computation volume becomes:

$$V_f = k(2v + 2mv) \tag{8}$$

If we enter an additional restriction on the type of trapezoidal membership functions, as shown in Figure 4, (i. e., to require that the slopes of the adjacent sides of the trapezoids are symmetric), then the values of μ_1, μ_2 will be mutually complementary to 1.

Then the number of multiplications will be halved, and the estimated amount of calculations becomes:

$$V_f = k(2v + mv) \tag{9}$$

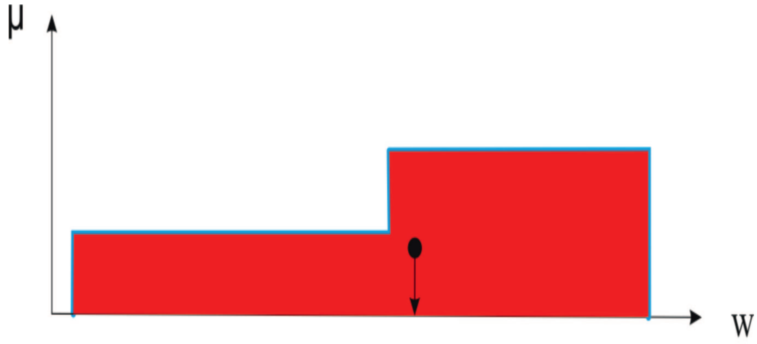


Figure 5: Illustration of the reduction of calculations in the stages of aggregation and defuzzification.

3.2 The method of simplified recording of the membership functions of the output variables and the implementation of the defuzzification stage

For this stage it is suggested to apply the interval assignment of the membership function of the output variable. As a result of aggregation, we get a union of rectangles, an example is shown in figure 5.

To obtain the output control signal with a fuzzy exit function, we use the criterion of the area's average. To do this, we calculate the area as the sum of products, divide by two by means of a shift operation and by iterative summation, with step Δw we determine the approximate center of mass and accordingly the clear value of the output function. Then

$$V_{df} = qv + nmv \tag{10}$$

where q- number of summation iterations with step Δw

Obviously, this method combines features of classical (with trapezoidal assignment of membership functions) and a simplified fuzzy inference algorithm, where as a result of processing the rules we obtain and process the constants Ci. It should be expected that the use of this method will support the "intelligence" of calculations close to the classical one, and save computing resources like a simplified algorithm.

Note that the above formulas reflect only approximate estimates since they do not take into account some other details of the operations performed in the microprocessor. So, we will conduct experimental studies and show that the calculated indicators of the reduction in the amount of calculations are manifested in a real reduction in the computation time.

At the same time, we will pay attention to changes in the quality control. By "quality" we mean the divergence of the values of the output control functions in comparison with some initial idealized model. The Mamdani model will be used as such a basic model in this paper.

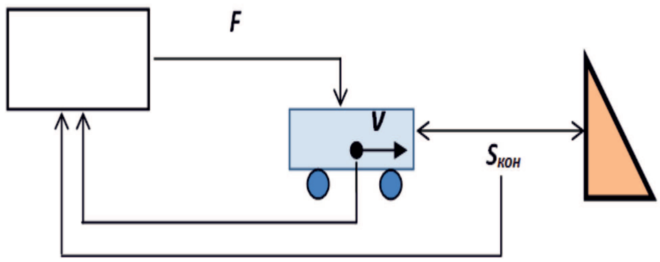


Figure 6: Mnemonic diagram of the test task for the development of the braking control algorithm of the mobile platform.

4 Research of the influence of methods of reducing computational complexity on the quality of control algorithms

As a test problem of fuzzy control we take the task of smoothly stopping the mobile platform before an obstacle, and as a control algorithm for braking we consider the classical method of the Mamdani algorithm. For example, the platform moves with the initial speed V_0, and at the distance S_0 from the obstacle the braking automat is activated. The platform should stop smoothly in the immediate vicinity of the obstacle. The braking control algorithm A continuously receives the current values of the speed v and the distance S_{con} remaining before the obstacle, and calculates the force F applied to the braking system (Figure 6).

It is required to realize the variants of the Mamdani algorithm (in the classical method and the proposed method of reduced calculations) and to conduct a series of model experiments. It is necessary to evaluate the quality of control and the time of calculations proposed in various ways during the experiments. As the initial data for the operation of the algorithm we set:

- input variables S_{con} and v;
- output variable F ;
- fuzzy inference rules.

Note that the input variables and fuzzy inference rules will be the same for the implementation methods being investigated and the output variable (according to B§ 3.2) will have differences. Thus, Figure 7 shows the membership functions of the input variables S_{con} and v. Figure 8 shows the functions of belonging to the given output variable F.

Next, the following fuzzy inference rules, where S_s, S_m, S_l are small, medium and large terms of distance, V_s, V_m, V_l are small, medium and large term of speed, F_s, F_m, F_l are small, medium and large term of the applied force, in the appropriate order:
If S_m and V_s then F_s,
If S_m and V_m then F_s,

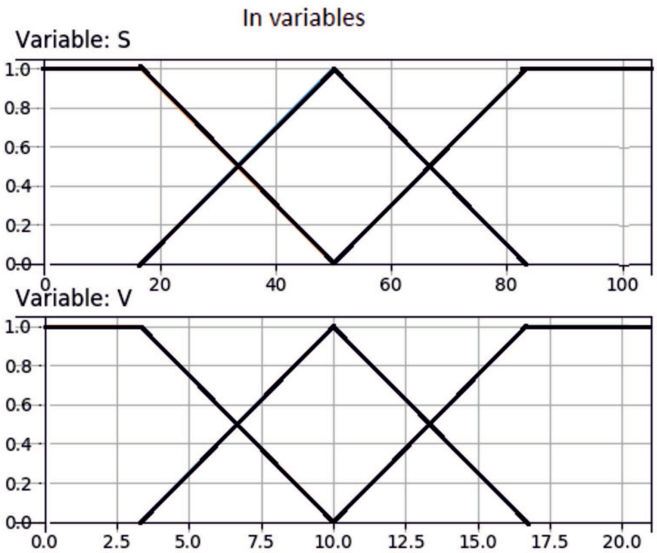


Figure 7: Functions of belonging to input variables.

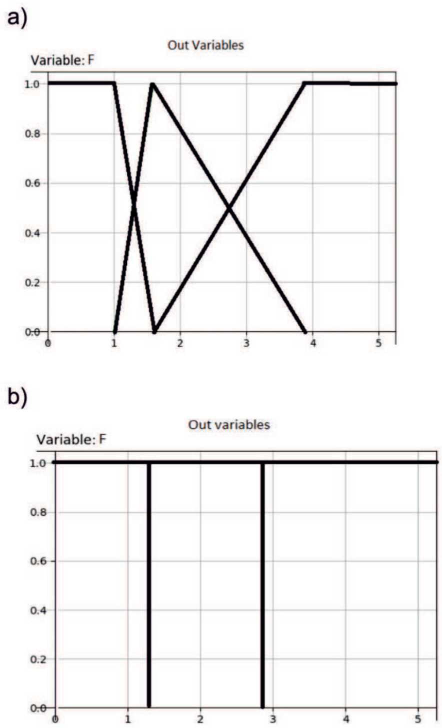


Figure 8: Functions of the output variable: in the classical method with a trapezoidal form - a; with a rectangular shape for the proposed method of reduced calculation вЂ“ b.

If S_m and V_l then F_m,
If S_l and V_s then F_s,
If S_l and V_m then F_s,
If S_l and V_l then F_s,
If S_s and V_s then F_m,
If S_s and V_m then F_l,
If S_s and V_l then F_l,

Modeling will be conducted according to the generalized algorithm for setting up experiments shown in Figure 9. In fact, all calculations associated with fuzzy inference occur in the "Calculation of effort" block. The block of tuning algorithms is intended for selecting fuzzy output rules and setting membership functions (in this article the algorithms of this block are not investigated). All other blocks serve to model the "external environment". We are to conduct three experiments.

In the first experiment we simulate the ideal case of inhibition, when the force F and the deceleration a are directly proportional, i.e. $F\alpha^{\circ}$. In this version it is assumed that the system can "provide" any necessary deceleration. Then for the calculation of the current values of v and S_con in the model of the "external environment" known formulas of slow motion are used:

$$v - v_0 - a\Delta t \tag{11}$$

$$\Delta x = v_0 \Delta t - a \Delta t^2/2 \tag{12}$$

$$x = x + \Delta x \tag{13}$$

$$S_{con} = S - x \tag{14}$$

In the second experiment we simulate the case of deceleration with a time lag in the actuation of the deceleration system's executive units.

In the third experiment we simulate the case of weak coupling of the platform with the road.

Visual analysis of simulation results is conveniently conducted on the basis of the dependence of the change in speed on time. So, in Figure 10 we can see a series of graphs, reflecting three different variants of braking started at the same speed. It is clear that with such characteristics of the movement of the platform there will be different distances to the stop. As can be seen, the first and second variants will differ by the smoothness of the braking, and the third graph illustrates the collision with the obstacle, i.e. the task is not executed. We will assume that there are experimental data on braking performed by an expert, while graph No. 2 reflects them. Let us take this dependence as the standard and adjust the fuzzy control algorithm. The achieved control quality will be considered as the proximity of the simulated braking modes to the standard one.

In the first experiment the above parameters of the input and output membership functions, as well as the product rules, allowed to obtain graphs of the classical and proposed implementation methods, practically merging with the standard graph. After

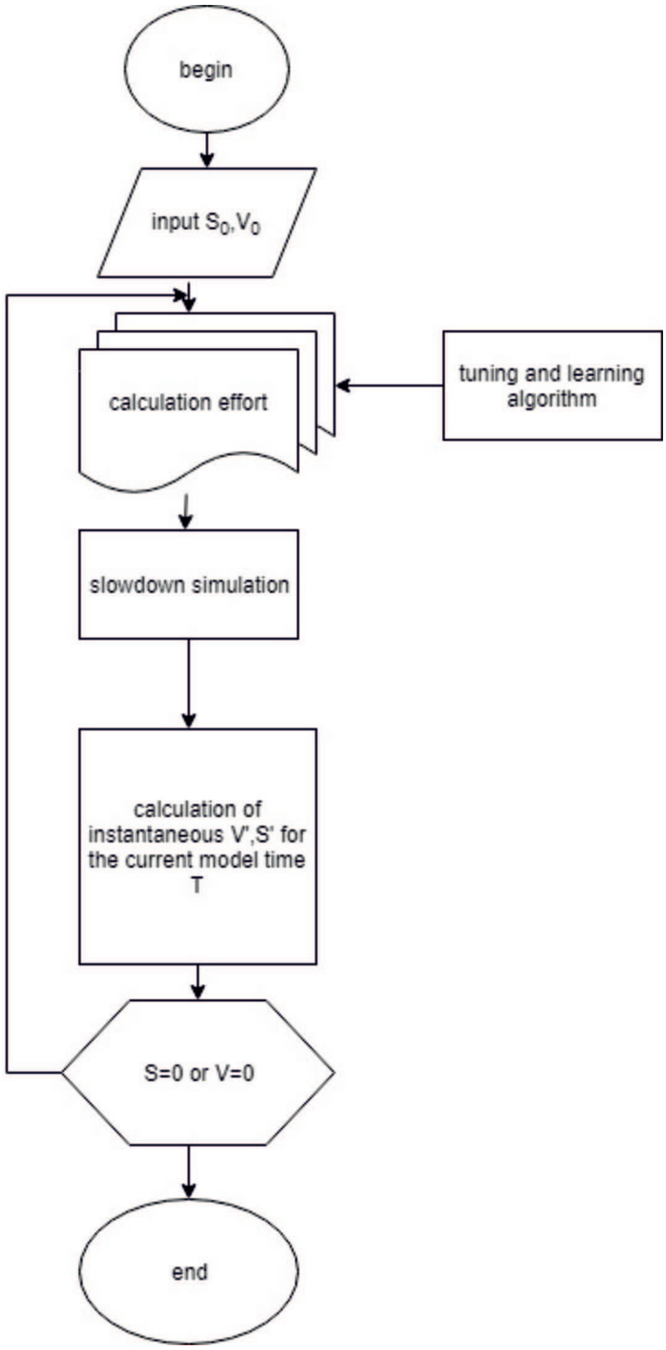


Figure 9: Block diagram of the general simulation algorithm.

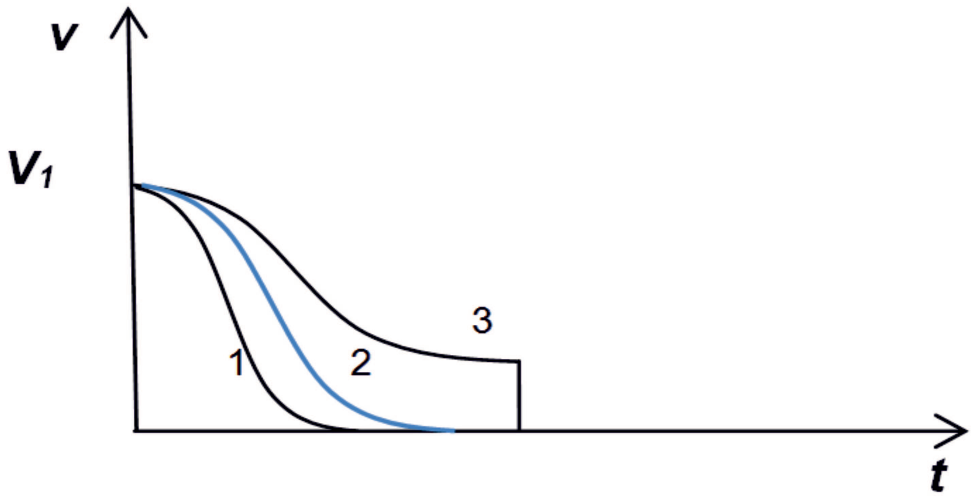


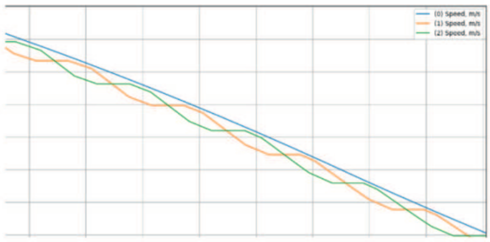
Figure 10: Graphs characterizing different dynamics of braking of the mobile platform.

the model was tuned and tested for the ideal case ($F\alpha a$), more complex variants of the influence of the environment on braking were investigated.

It is worth recalling that in real conditions such factors as the inertia (mass) of the platform, the percentage of slope, the coefficient of adhesion to the surface, the response time of the braking system actuators, etc. influence the deceleration. However, in the above model, the named parameters are not contained. Nevertheless, there is the possibility of an indirect imitation of the influencing factors on the deceleration, for example, by inletting the delay time in the braking algorithm, etc. Thus, a control signal queue was applied in order to simulate the delay of the actuating devices of the braking system. The output of the control signal queue was accompanied by a delay, which allowed simulating the delay between the output of the signal and the actuation of actuators in the real device. There was a limit on the maximum value of the output control signal for deceleration F in order to simulate a low (zero) coefficient of traction with the road. This allowed modeling of extreme road conditions (icing, wetting, spilled oil, etc.).

After applying the limit on the maximum value, the addition of a certain value of O"F to the control signal for deceleration was applied to simulate slopes, wind, etc. For such sophisticated models of the external environment, control algorithms should exhibit adaptability, i.e. strive to "correctly" stop the platform despite the negative impact of external factors. Just in these cases, the strengths and weaknesses of the algorithms being developed are revealed.

In the second and third experiments, the influence of these factors on the braking process and the efficiency of the control algorithms under such conditions were investigated. In the course of the experiments, the results of modeling the two realizations (classical and developed) of the braking control algorithm are obtained, when: a) the delay in the operation of the actuators is observed in the system; b) in some area there is no cohesion of the wheels with the road. Figure 11 shows fragments of the graphs of the velocity versus time dependences. The idealized form of the graph is colored a



b

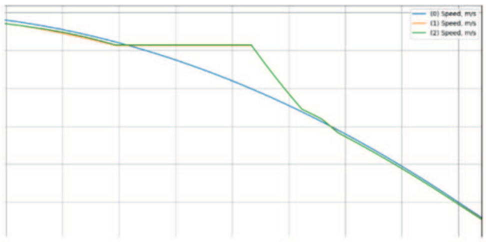


Figure 11: a fragment of the curves for modeling the braking dynamics: a) with a delay in the actuation of the actuators; b) a section with a reduced coefficient of adhesion to the road.

blue, green is the implementation of the classical method, the implementation of the proposed method with reduced calculations colored orange. In the first variant we see that the algorithms are delayed but they work out the general dynamics of speed reduction. In the second variant we observe the absence of a decrease in speed on a slippery section of the road, and then the "aspiration" of the algorithm to accelerate the deceleration.

It is also clear from the figure that the graphs of the classical and simplified methods of the Mamdani algorithm practically merge with each other. This indicates that the quality of the fuzzy controller algorithms with a simplified implementation does not deteriorate.

To estimate the speed of the algorithm, we calculated the average execution time based on the results of 10 runs of each of the algorithm variants with the same input data. The fuzzification in the simulation cycle was performed in three terms for each of the input variables S and V, with trapezoidal (triangular) membership functions. The number of terms of the output variable F in both experiments was also equal to 4, but in the classical implementation the membership function was trapezoidal, and the interval function was in the simplified implementation (see $3).

Only simulation time was taken into account to avoid distortion of results due to the influence of overhead on program initialization. The results of the experiments are summarized in Table 1. As follows from the obtained results: the application of the proposed methods of reducing the computational complexity of the algorithm will improve the system performance by almost 8 times.

	The classical method of implementation	The simplified method of implementation (proposed)
Machine time, sec	9.5	1.2

4 Conclusion

In this article we have suggested the most effective ways to reduce the complexity of computing fuzzy inference algorithms, which do not lead to a valuable decrease in the control quality. By the quality of management we understand the overall speed of the control system, the smoothness of the change in the regulated parameter, the resistance to noise and fluctuations, adaptability, learnability, etc. During the research of fuzzy inference algorithms with minimization of computational complexity, the following problems were solved: the evaluation of the computational efficiency at the stages of fuzzy inference was performed; methods for reducing computational complexity by simplifying computations at a lower algorithmic level were suggested. The experimental studies have been conducted, as a result of which it was demonstrated that the calculated indicators of the reduction in the amount of calculations were manifested in a real reduction in the computation time. The simplified model of fuzzy inference proposed by us was compared with the classical implementation of the Mamdani algorithm, and as a test problem of fuzzy control, the problem of smooth stopping the mobile platform before the obstacle was chosen. Also, to visualize the results, we provided a series of graphs reflecting three different variants of braking started at the same speed (idealized, classic and simplified). As a result, there was no discrepancy between the values of the simplified output control functions in comparison with the classical model. This means that the quality of the fuzzy controller algorithms with a simplified implementation does not deteriorate. The application of our algorithm for fuzzy inference will allow us to use inexpensive low-power controllers, which will reduce the cost and increase the reliability of the final system.

References

[1] Krabs, Werner, Zowe, Jochem (Eds.), *Modern Methods of Optimization, Proceedings of the Summer School вЂњModern Methods of OptimizationвЂќ*, held at the SchloГџ Thurnau of the University of Bayreuth, Bayreuth, FRG, October 1вЂ“6, 1990, P. 351

[2] F. Rahnman, et al., *Network Approach for Physiological Parameters Measurement*, IEEE Trans. on Instrumentation and Measurement, Vol. 54, No. 1, Feb. 2005, pp. 337-346

[3] E. TГіth-Laufer, A. R. VГЎrkonyi-KГіczy, *A Soft Computing-based Hierarchical Sport Activity Risk Level Calculation Model for Supporting Home Exercises*, IEEE Trans. on Instrumentation and Measurement, Vol. 63, No. 6, 2014, pp. 1400-1411, doi: 10.1109/TIM.2014.2299523

[4] Y. Koyama, M. Nishiyama, K. Watanbe, *вЂњA Motion Monitor Using HeteroCore Optical Fiber Sensors Sewed in Sportswear to Trace Trunk MotionвЂќ* , IEEE Trans. on Instrumentation and Measurement, Vol. 62, No. 4, Apr. 2013, pp. 828-836, doi: 10.1109/TIM.2013.2241534

[5] G. Hache, E. D. Lemaire, N. Baddour, *Wearable Mobility Monitoring Using a Multimedia Smartphone Platform* , IEEE Trans. on Instrumentation and Measurement, Vol. 60, No. 9, Sept. 2011, pp. 3153- 3161, doi: 10.1109/TIM.2011.2122490

[6] Zi-Xing Cai , *Series in Intelligent Control and Intelligent Automation*, Intelligent Control: Principles, Techniques and Applications, Volume 7

[7] G. Saini , *Application of fuzzy logic to real-time scheduling*, 14th IEEE-NPSS Real Time Conference, 2005. Stockholm, Sweden, 12 December 2005, 1875 p.

[8] Mojtaba Sabeghi,вЂ and Mahmoud Naghibzadeh , *Fuzzy Algorithm for Real-Time Scheduling of Soft Periodic Tasks* , International Journal of Computer Science and Network Security, VOL.6 No.2A, February 2006, P. 227

[9] Wolfgang Rankl,Wolfgang Effing , *Smart Card Handbook*, , Wiley; 4 edition (July 19, 2010), P.1088

[10] Zadeh, L.A. , *Fuzzy sets, Information and Control*, Vol. 8, No. 3, pp. 339-353, June 1965, ISSN: 00199958

[11] Zadeh, L.A. , *The concept of a linguistic variable and its application to approximate reasoning* , part I. Inf. Sci. 7(1), 199вЂ“249 (1975)

[12] Jang Jyh-Shing Roger, *Neuro-fuzzy And Soft Computing: A Computational Approach To Learning And Machine Intelligence* , (18-Jun-1905), Paperback, 25-26 p.

[13] Cox, E., *The Fuzzy System Handbook: A Practitioner's Guide to Building* , Using, and MaintainingFuzzy Systems, Boston: Academic Press, 1994, 365 p.

[14] Mamdani, E. H. , *A Novel Adaptive Fuzzy Controller Approach of Brushless DC Motors without Hall and Position Sensors, PrzeglД. . . d Elektrotechniczny, 12a/2012 Page no. 290*, IEEE Transactions on Computers 26(12)

[15] Xia Mingzi , Liu Yang , Zhao Jin , *A Novel Adaptive Fuzzy Controller Approach of Brushless DC Motors without Hall and Position Sensors*, Xia Mingzi , Liu Yang , Zhao Jin

Zhartybayeva Makpal Galymbekovna,
L.Gumilyov Eurasian National University,
Pushkin st. 11, Faculty of information Technologies, Astana, Kazakhstan,
Email: makkenskii@mail.ru,
Tatur Mikhail Mikhailovich,
Belarusian State University of Informatics and Radioelectronics,
Minsk, Belarus, Department of Electronic Computing Machines,
Email: tatur@i-proc.com,
Shaverdo Mikhail Mikhailovich,
Belarusian State University of Informatics and Radioelectronics,
Department of Electronic Computing Machines, Minsk, Belarus,
Email: m.shaverdo@gmail.com,
Iskakov Kazizat Takuadinovich,
L.Gumilyov Eurasian National University,
Pushkin st. 11, Faculty of information Technologies, Astana, Kazakhstan,
Email: kazizat@mail.ru.

Received 24.11.2018, Accepted 10.01.2019.