Advances in Science and Technology Research Journal

Advances in Science and Technology Research Journal 2023, 17(5), 374–384
https://doi.org/10.12913/22998624/171810
ISSN 2299-8624, License CC-BY 4.0

Received: 2023.07.07
Accepted: 2023.09.04
Published: 2023.10.20

Implementation of Type-2 Fuzzy Controller in Matlab Software

Ireneusz Dominik[1*], Stanisław Flaga[1]

[1] Department of Process Control, Faculty of Mechanical Engineering and Robotics AGH University of Krakow, Al. Mickiewicza 30, 30-059 Kraków, Poland

* Corresponding author's e-mail: dominik@agh.edu.pl

ABSTRACT

The purpose of this work is to create a Matlab toolbox that makes it easy and accessible to become acquainted with a novel control method called type-2 fuzzy controller. A toolbox for working with type-1 controllers can be found in the Simulink package, while there are only few, simple toolboxes for type-2 fuzzy controllers. The article describes the details of the created software, which allows working both with simulation objects, but also enables creating program code for a PLC industrial controller. This gives the opportunity to work in a simulation environment with a model of the control object and then, after tuning the controller, to automatically implement the controller to control the real object. In the literature, one can find many methods for reducing type-2 to type-1 fuzzy logic, but most often they are compared to several well-known classical reduction methods, such as the KM algorithm. There is no compilation of the most popular methods and a comparison of their performance. With the new toolbox it was possible to quickly create and add new reduction methods; thus, an analysis of 16 reduction methods was also presented in the article.

Keywords: type-2 fuzzy controller, toolbox, reduction types, Matlab.

INTRODUCTION

Continuous advances in control theory have led to the need to reach for unconventional solutions. Work striving for better controllers has developed an idea called fuzzy logic. Although it was created as early as the 1960s [1], its practical application development has occurred only recently [2, 3]. The use of fuzzy controllers is becoming popular, due to the ease of implementation and the good quality of regulation [4]. At the turn of the last few years one can notice an increase in interest in type-2 fuzzy logic, which is a development of the classical logic now called type-1 [5]. In type-2 fuzzy logic we operate on three-dimensional membership functions, which, on the one hand, expands the applicability of such applications, on the other hand, unfortunately, makes calculations more difficult [6]. In addition, despite the creation of several tools for working with such logic, there is still a need to build a universal toolbox easily accessible to researchers or engineers.

In the case of fuzzy logic type-1, a given element x is characterized by the degree of membership in the fuzzy set, which is a real number, while in the case of fuzzy logic type-2, the degree of membership is fuzzy in nature. Thus, it is possible to reduce the inaccuracy of the operation of the control system. This inaccuracy can arise from the differing opinions of experts. Their knowledge and conclusions (which form the rule base) are different because the same linguistic terms can be defined differently by different experts [7, 8]. In addition, type-2 fuzzy logic is far more effective in accounting for uncertainty [9].

Another distinguishing feature of type-2 fuzzy sets is the addition of a third dimension forming a secondary membership function. If this function has a rectangular shape and takes a value equal to 1 for all points of the "base" called primary membership function, we speak of interval type

2 fuzzy sets IT2 FS [8]. However, if the secondary membership function has the shape of, for example, a triangle, then its values are different for different points of the primary membership function, and then we speak of general type 2 fuzzy sets GT2 FS (Fig. 1). According to Mendel's translation [8], the primary membership function is formed by a group of the same type 1 fuzzy functions placed next to each other, over which the secondary membership function determines the probability of selecting a given type 1 function from among the whole group. The secondary membership function is thus a kind of weight for the primary membership function.

The control system when using type-2 fuzzy logic (Fig. 2) is not significantly different from the type-1 fuzzy system [10].

The main difference marked in Figure 2 by the dashed line is the output processing. Defuzzyfication of fuzzy sets is the most processor-intensive operation of the type-2 fuzzy controller. It consists of two stages. Firstly, it is necessary to perform the so-called type reduction which transforms a type-2 fuzzy set into a type-1 fuzzy set. The resulting set is called a centroid, and it can be further processed to a crisp value, which is the value of the controller's output signal. An extensive study on the performance of reduction algorithms is presented in section 4.

Several toolboxes for working with type-2 controllers are described in the literature, they are built not only in Matlab software [11, 12] but also with the usage of different languages e.g. Java [13, 14], R language [15] or Python [16]. However, they are characterized by limitations in applications, the vast majority are limited to interval controllers, or without the possibility of any significant changes int the controller's structure.

A comparison of the author's toolbox called AGH FLS2 described in the article with two commonly used toolboxes is shown in Table 1.

Type-2 fuzzy controllers are still little known, with few scientists and even fewer engineers having knowledge related to them. One limitation is the limited ability to work in a simulation environment, where new control algorithms can be understood and practiced without additional costs or expensive equipment. In addition, important

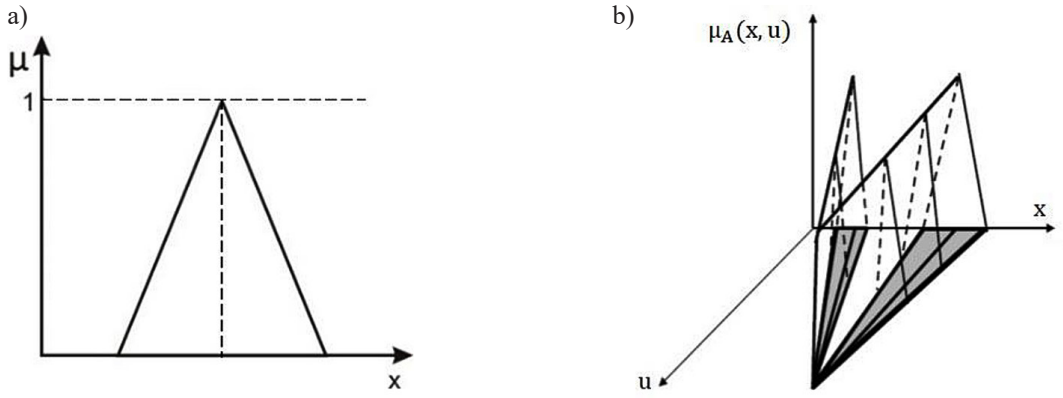

Fig. 1. The difference between fuzzy logic membership function: a) type-1, b) general type-2

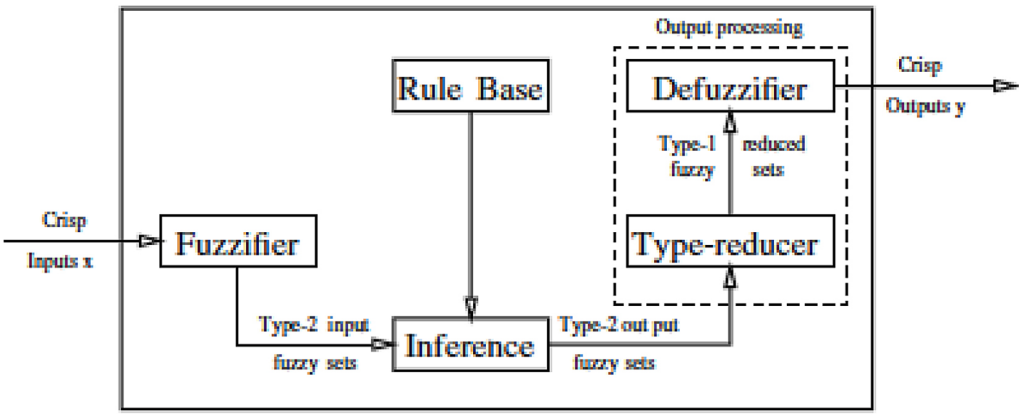

Fig. 2. Type-2 fuzzy logic system structure

Table 1. Comparison of the author's toolbox AGH FLS2 with other toolboxes

Feature compared		*AGH FLS 2*	*IT2 FLS Toolbox* [17]	*Juzzy* [18]
Supported fuzzy sets	IT2FS MF	gauss, gbel, sig, s, trap, tri, z, any of the user	gauss, gbel, pi, sig, dsig, psig, s, tri, trap, z	tri, trap, gauss, gau-angle
	GT2FS	zSlice (any number of planes)	no	zSlice (4 planes)
Defuzzification	available algorithms	KM, EKM, IASC, EIASC, EODS, WM, NT, BMM, CJ, DY, EODS, G, LM, LYZ, NT, TTCC	KM, EKM, IASC, EIASC, EODS, WM, NT, BMM	KM
	methods	centroid, COS, CSUM, Height,	no choice	Centroid, Height

especially for engineers, is the need to migrate from the simulation environment more easily to the ability to develop practical applications. Such an option is not actually found in any of the toolboxes currently available. Therefore, the AGH FLS 2 toolbox described in the article not only allows much more freedom in creating a controller by developing your own membership functions or reduction methods, but also allows automatic code generation for a real PLC and immediate use of it for controlling a real object.

The AGH FLS2 toolbox in the Matlab/Simulink environment

The toolbox was created in two stages. In the first, the focus was on the implementation of interval membership functions, as is the case with almost all other toolboxes, and then, after a year testing, the possibility of using general membership functions was added, in which the Z-plane method (zSlices) was used for calculations.

It was assumed that in research activities there should be no restrictions on the number of membership functions and the shape of the functions. In addition, it was assumed the need for extensive visualization for general functions as well, and the ability to test and create own reduction methods. The result of such assumptions was the toolbox the main window of which is shown in Figure 3.

The main application window consists of blocks containing Fuzzy Inference System (FIS) setting options: product method (min or prod), sum method (max or probor), implication method of sets (min or prod), aggregation method of sets (max or probor), reduction method (16 available), defuzzification (centroid, cos, csum, height, md_ height). Additionally, selection of membership function of general or interval type is also available.

In the toolbox, unlike those presented earlier, there is a choice of nine symmetric and asymmetric membership functions: triangular, trapezoidal,

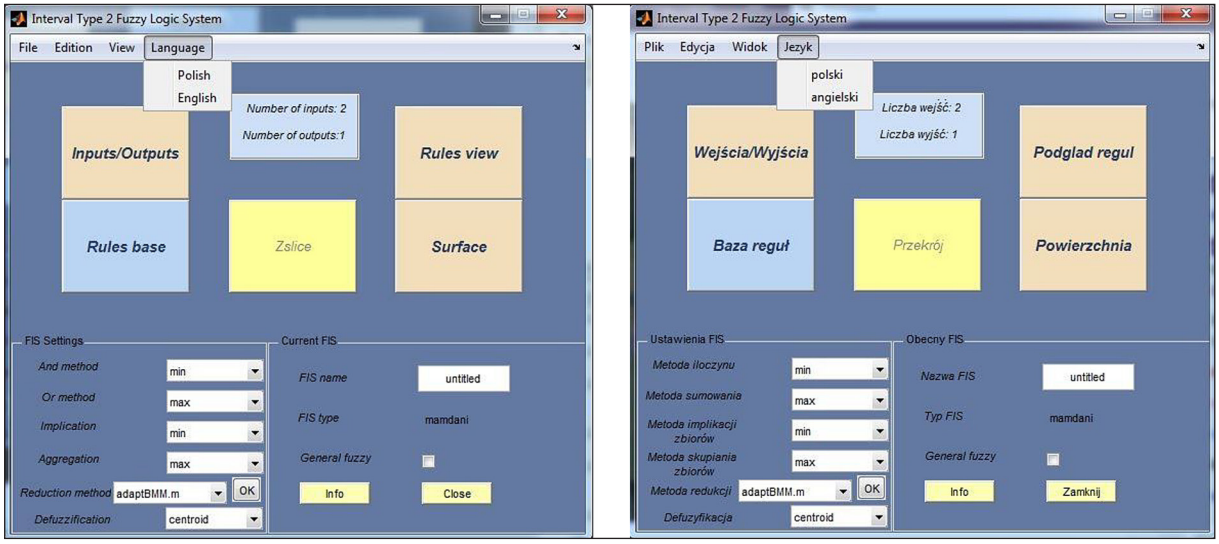

Fig. 3. The appearance of the main window of the toolbox in English and Polish

Gaussian, bell, S-type and Z-type. The files used to generate type-2 membership functions can be used independently, outside the IT2FLS software, or new function types can be created based on them (Fig. 4).

In the editing window input/output the membership function parameters with their visualization is available (Fig. 5).

The Zslice selection slider gives the ability to divide the secondary membership function at any level and is not limited to integers. This ability is useful when you want to have a visualization of the cross section of the function at the selected level Z (Z∈N). Other windows similar to FLS-1 toolbox are also available e.g.: rule base or control surface visualization.

The original functionality of the toolbox is the ability to add defuzzification methods. It creates a file with the new defuzzification method which can be used both in the toolbox and apart from it.

AUTOMATIC GENERATION OF TYPE 2 FUZZY CONTROLLER PROGRAM CODE TO B&R PLC FROM MATLAB/ SIMULINK ENVIRONMENT

The ability to automatically generate program code is offered by Bernecker&Rainer, a company specializing in industrial automation, which provides the Automation Studio Target for Simulink ASTfS add-on. The Simulink tool allows the

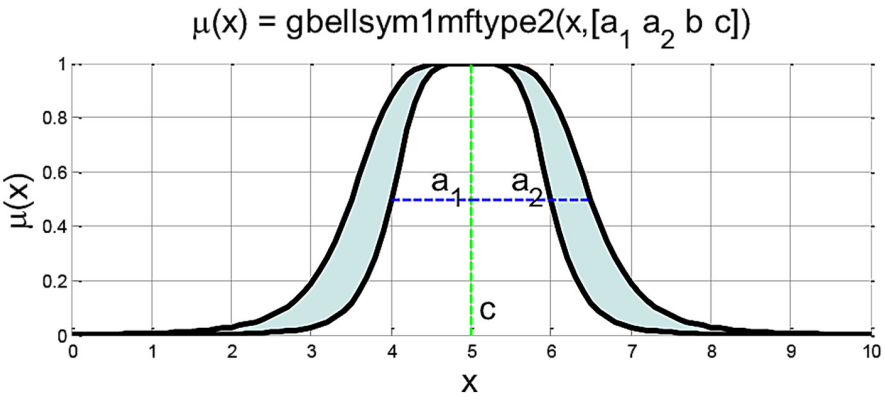

Fig. 4. Bell symmetric function gbellsym described by four parameters

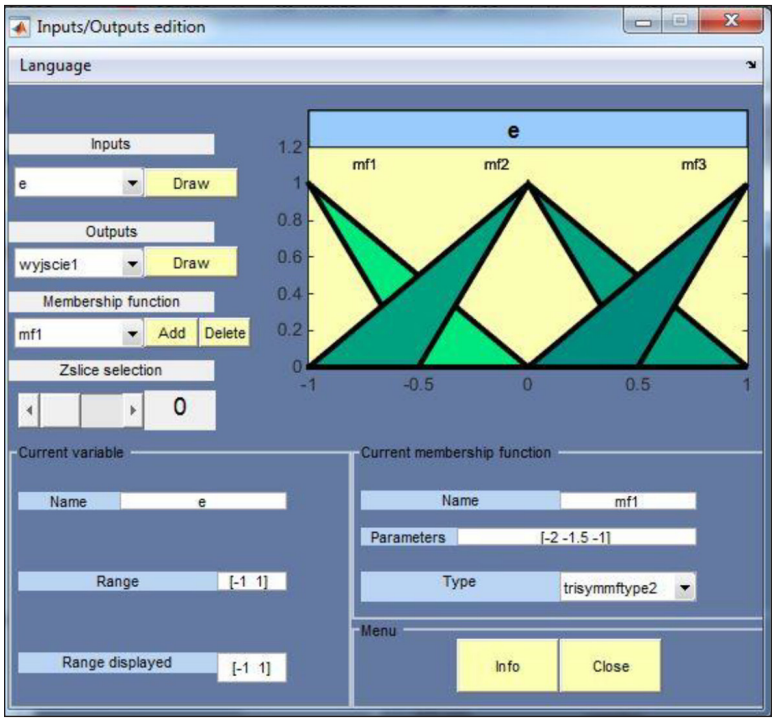

Fig. 5. Input/Output MF editing window

creation of advanced control system schematics, while the ASTfS add-on allows the control system schematic to be converted to C/C++ language and used in a B&R PLC (Automatic Code Generation). A controller programmed in this way can immediately control a real object, without additional intervention in the controller program and time-consuming controller parameterization, it is a software variation of rapid prototyping. The risk of errors in the program code is also reduced, due to the elimination of the human factor. Automatic code generation significantly reduces the cost and time of developing control systems.

Conversion of a type-2 fuzzy controller program and object model from Matlab/ Simulink environment to C language

In order to use the type-2 fuzzy controller structure created in the AGH FLS2 toolbox for automatic code generation, an existing type-1 fuzzy controller file in Matlab written in ANSI C was modified. This use of the modified controller allows the use of the standard Fuzzy Logic Controller block from the library in Simulink to work with the type-2 controller structure. In addition, the block automatically recognizes which type of controller it is dealing with, based on the structure sent to it.

In order to convert the controller structure, it was decided to create its own System-function sffis file. To check which way is currently used one can select *Look Under Mask* option (Fig. 6).

The S-function is a tool that enhances the capabilities of the Simulink environment. It can be created in C/C++, Matlab or Fortran languages. With S-functions, it is possible to create new blocks in Simulink, use previously written algorithms in one of the mentioned languages or significantly speed up simulations. Modifications

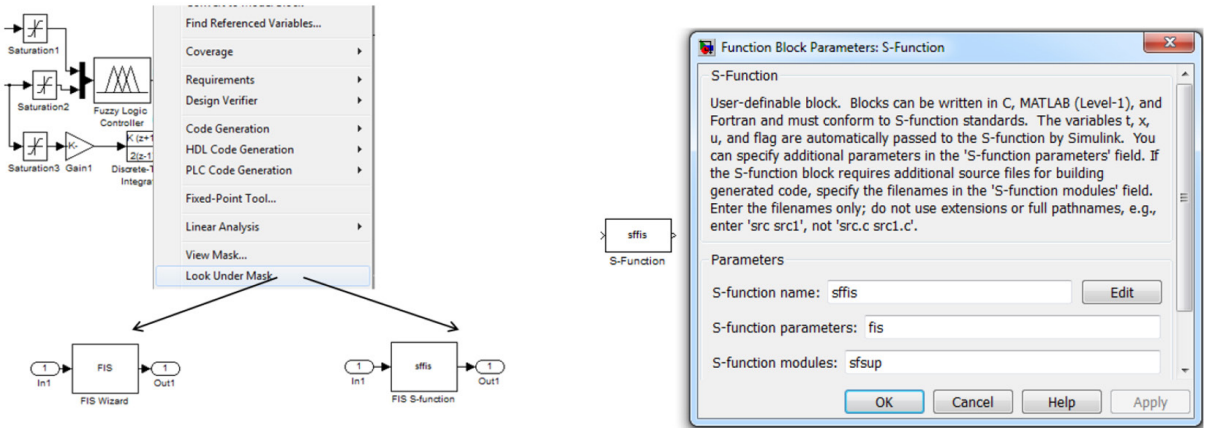

Fig. 6. Checking the calculation method used in the Fuzzy Logic Controller block and the properties window of the S-Function block

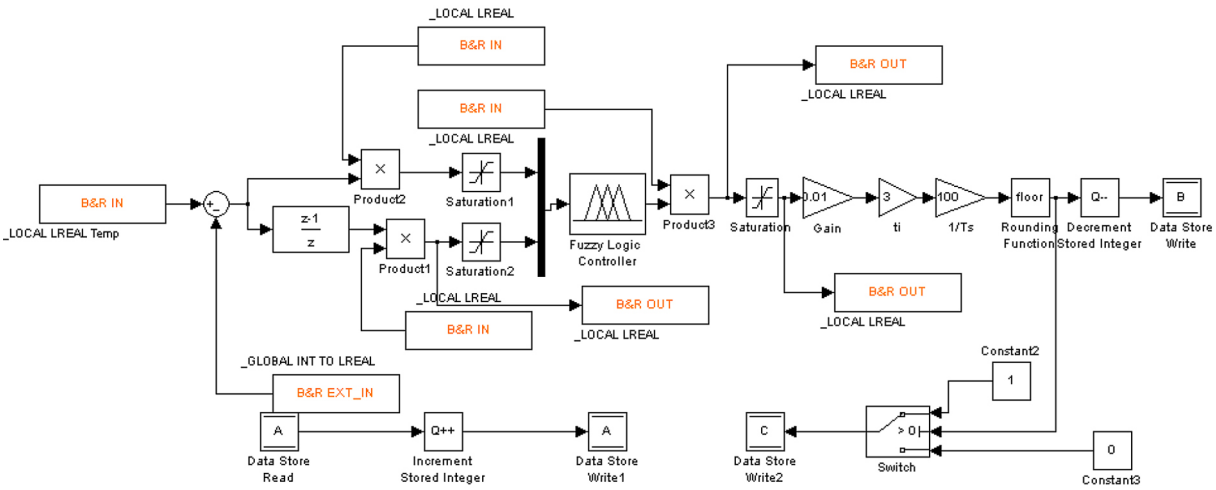

Fig. 7. Example of control system in Simulink with additional B&R blocks

to the controller were made on Matlab source files. The files *fis.h* and *sfsup.c*, responsible for the operation of the controller in Simulink, were changed. Additionally, the file *sffis.tlc* was also significantly modified, its task is to create the structure of the controller in ANSI C language during code generation. In addition to performing calculations for the type-2 fuzzy controller, the modification allows for a thorough analysis of the inference stage.

When the fuzzy controller is run in Simulink, its structure from Workspace is converted into an ANSI C language structure using the *matlab2cStr* function. A single button press is sufficient for the controller's code in Simulink to be generated, compiled and sent to a PLC (Fig. 7).

This procedure is sufficient for simple control objects, while for advanced fuzzy controllers, some compiler files must be modified additionally. A fragment of the generated controller structure using the modified *sffis.tlc* file is shown in Figure 8.

Once the code has been generated in Simulink and uploaded to B&R Automation Studio, the *fis.h* and *sfsup.c* files must be manually added to the project tree. Then one has to open the

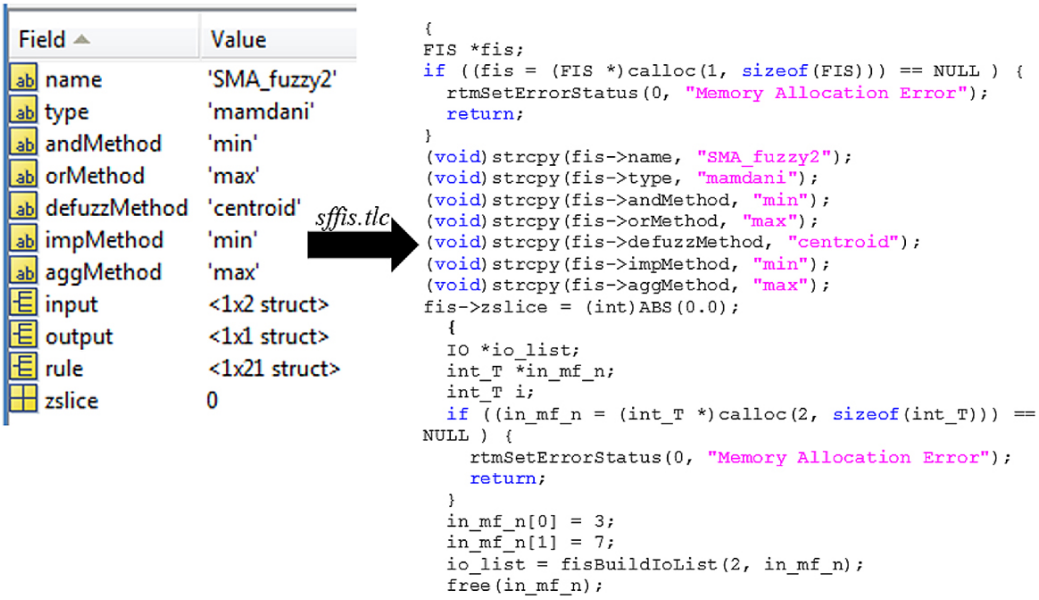

Fig. 8. An excerpt of a C-language controller structure generated from Simulink using a modified *sffis.tlc* file

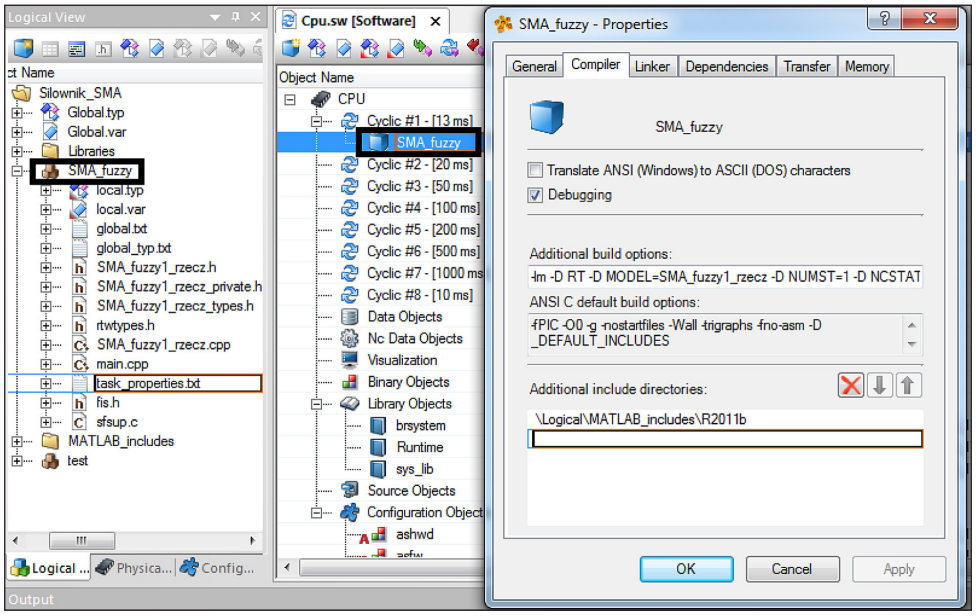

Fig. 9. View of the program tree in B&R software generated automatically from the block diagram in Simulink

Compiler tab and complete the compilation options, as shown in Figure 9. After the changes, the program compiles without errors and can be uploaded to a PLC. A sample project "SMA_fuzzy" created in Simulink was transferred to B&R Automation Studio, the code with local and global variables as well as a cyclic task realized every 13s is visible in Figure 9.

PLCs are usually programmed using IEC 61131-3-compliant languages, e.g. ladder diagram or structured text. The controller structure file obtained from the Matlab environment is created in C language using B&R software - Automation Studio Target for Simulink ASTfS add-on, and by default can only be used in B&R PLCs. It is also possible to try to implement the controller in PLCs that support the C language, such as the new versions of the S7-1500 series from Siemens. However, this would require at this point manual modification of the resulting code to match the given controller.

ANALYSIS OF REDUCTION METHODS

In the literature [19, 20], one can find many methods of reducing type 2 to type 1, but most often they are compared to several well-known classical reduction methods, such as the Karnik-Mendel KM algorithm [21], while there is no compilation of the most popular methods and comparison of their performance. Because of its crucial role in the type-2 fuzzy controller performance all the time new reduction methods are researched [22, 23].

With access to a toolbox that allows the rapid creation and usage of new reduction methods, such an analysis was conducted. For the purpose of the reduction study, an add-on was created in the toolbox, which allows the user to choose out from the commonly used sixteen reduction methods.

The most important parameter of type reduction algorithms is the accuracy of calculations, but even the best method that gives very good reduction results cannot be used in the control of real-time systems if it too slow. For the selected fuzzy set, the computation time of individual reduction algorithms was defined .The calculation time was measured by *Run and Time* option available in Simulink. The computer specification used in research was as followed: Intel Core i5 Quad-core processor (Intel 11th Generation 2.4GHz), 24 GB RAM (DDR4 SDRAM) and 500 GB SSD.

Simulation results for all reduction methods are summarized in Table 2. The reduction was carried out for the same single membership function with the Bell symmetric function shape for all tested methods. The output values obtained from the calculations are also compared.

In fuzzy control systems based on the Mamdani model, the output value of the controller is an interval function of the type-2 fuzzy set, which is reduced to type-1 reduced fuzzy set and ultimately to a single numeric value (Figure 2). Most often to reduce the type Karnik-Mendel (KM) iterative

Table 2. Summary of simulation results of reduction methods

No.	Reduction method	Calculation time [ms]	Result of reduction [–]
1	Karnik–Mendel algorithm	50	0.7188
2	Enhanced Karnik–Mendel algorithm	26	0.7235
3	Enhanced Karnik–Mendel algorithm 2	36	0.7188
4	Iterative algorithm with stop Condition	30	0.7188
5	Enhanced iterative algorithm with stop condition	26	0.7331
6	Enhanced iterative algorithm with stop condition 2	30	0.7188
7	Enhanced opposite direction searching algorithm	36	0.7188
8	Nie–Tan method	18	0.7153
9	Begain–Melek–Mendel method	20	0.7235
10	Wu–Mendel method	26	0.7135
11	Liang–Mendel method	22	0.7235
12	Coupland–John method	42	0.6806
13	Gorzalczany method	20	0.8300
14	Li–Yi–Zhao method	20	0.7930
15	Du–Ying method	20	0.7153
16	Tao–Taur–Chang–Chang method	28	0.7191

algorithm is used [6]. At the first step of the algorithm's implementation, it is necessary to determine the average initial set, which is the arithmetic mean of the upper and lower initial sets. In the next step, a new embedded set is determined such that for values smaller than the determined center of gravity it takes on values equal to the lower membership function, while for larger values it takes on values equal to the upper membership function. If the determined center of gravity in a given iteration is different from the previously determined one, the operation described previously is repeated until the step in which the determined center of gravity equals the value determined in the earlier step. The iterative methods are associated with a large computational effort, so many alternative methods have been developed. Generally, in these methods, the value of the output is determined using a different formula's proposed by scientist. The details of the reduction are available e.g. in [7, 8, 23].

From the summary, it can be read that the Karnik-Mendel method (1) has the longest calculation time. However, it is accurate and widely used, so the other methods were compared to it.

Among all the iterative methods discussed, the Enhanced Karnik-Mendel algorithm (2) turned out to be the best, mainly due to its significant time reduction. In the case of non-iterative reduction methods, the best results were given by the Nie-Tan method (8). However, it should be kept in mind that its efficiency is due to the simplicity of the calculations and for complex membership functions only the approximate output value of the reduction result can be obtained.

In further research, the operation of a type-2 fuzzy controller was simulated using various type-reduction methods. An oscillating object model was used in the control system, its transfer is presented in the formula 1:

$$G(s) = \frac{1}{s^2 + s + 1} \tag{1}$$

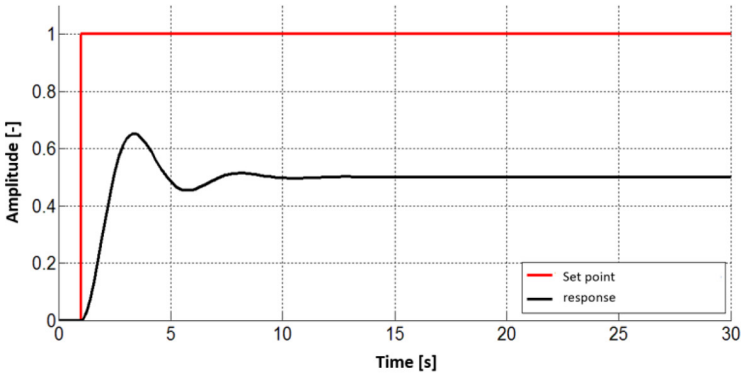

Fig. 10. Step response of the oscillating object given by formula 1

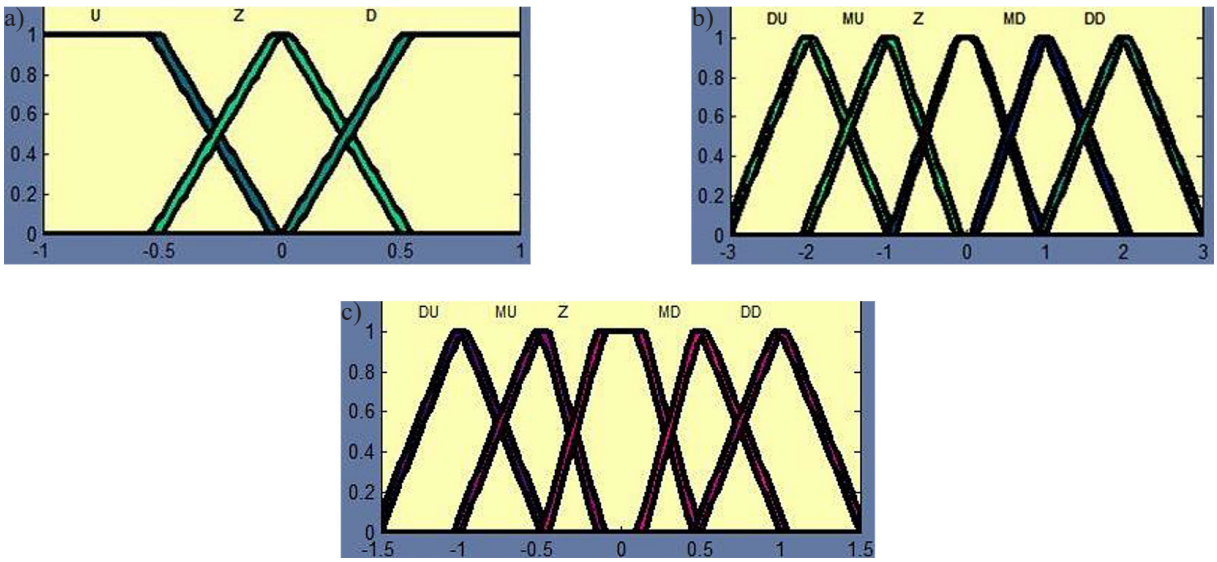

Fig. 11. Distribution of the membership function of a type-2 controler: a) error, b) error integral, c) output

Step response is shown in Figure 10. From the plot, it can be read that the steady-state error is 0.5, which is half of the setpoint.

Next, the type-2 fuzzy controller was designed in the toolbox. Two inputs were assumed: error and error integral. Fuzzification was carried out using the membership functions with the designations: DU – large negative, MU – small negative, Z – zero, MD – small positive, DD – large positive (Fig. 11).

In the next step simulation studies were conducted for the oscillating object described by equation (1). Simulation data were collected for all 16 reduction methods and were summarized in Table 2. Figure 12 shows exemplary step responses for selected methods.

All Karnik-Mendel type reduction algorithms have identical responses with reduced computation time for KM-enhanced methods. From the characteristics comparing the results of simulations using the Li-Yi-Zhao [24] and Coupland-John algorithms [25], the first method gives exactly the same response as KM. The second algorithm gives significant deterioration in control quality (Fig. 12). Table 3 summarizes the control performance criteria for the discussed reduction methods: response time – the time required for the response curve to reach and stay within a range of 2% percentage of the final value, steady-state error – the difference between the desired final output and the actual one, overshoot – the maximum peak value of the response

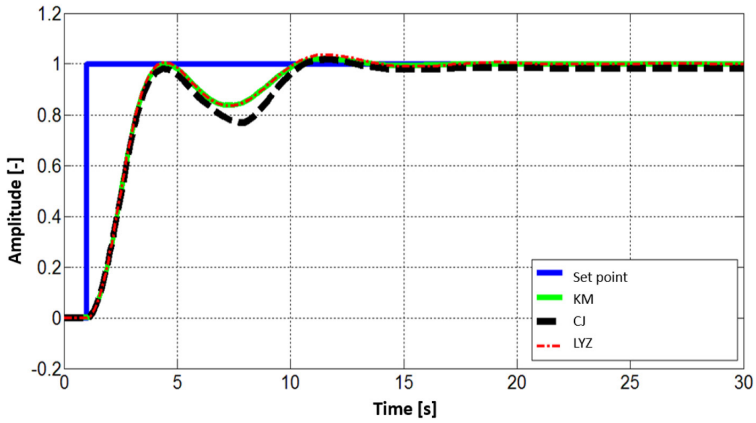

Fig. 12. Selected graphs comparing the performance of type reduction algorithms: KM -Karnik-Mendel (1), CJ -, Coupland–John (12), LYZ - Li–Yi–Zhao (14)

Table 3. Summary of performance of the controller quality for the oscillating object model

No.	Reduction method	response time [s]	steady-state error [–]	overshoot [%]	integral criterion IAE
1	Karnik–Mendel Algorithm	9.43	0	2.15	18.16
2	Enhanced Karnik–Mendel Algorithm	9.43	0	2.21	17.32
3	Enhanced Karnik–Mendel Algorithm 2	9.43	0	2.18	18.10
4	Iterative Algorithm with Stop Condition	9.43	0	2.19	17.32
5	Enhanced Iterative Algorithm with Stop Condition	11.6	0	4.6	32.59
6	Enhanced Iterative Algorithm with Stop Condition 2	9.43	0	2.17	17.91
7	Enhanced Opposite Direction Searching Algorithm	9.43	0	2.18	18.43
8	Nie–Tan Method	9.44	0	2.16	16.40
9	Begain–Melek–Mendel Method	9.43	0	2.2	17.32
10	Wu–Mendel Method	9.44	0	1.97	16.32
11	Liang–Mendel Method	9.43	0	2.2	17.32
12	Coupland–John Method	9.87	0.017	1.88	29.50
13	Gorzalczany Method	9.86	0.016	1.87	28.47
14	Li–Yi–Zhao Method	9.42	0	3.69	17.92
15	Du–Ying Method	9.43	0	2.16	16.42
16	Tao–Taur–Chang–Chang Method	9.43	0	2.18	18.59

curve measured from the desired response of the system, IAE Integral Absolute Error - integrates the absolute error over time.

Most of the simulated algorithms gave very similar responses. Only three methods gave slight discrepancies. The calculation time of each method was also simulated. This is a very important parameter since it determines whether a particular reduction method can be applied to a real system. The Wu-Mendel and Nie-Tan method turned out to be the most effective for the researched oscillating object.

Additional studies related to type reduction have been carried out for a model with the transfer function of a first-order model with delay. However, only a few methods were compared, primarily iterative methods based on the Karnik-Mendel model (1, 2, 3) and Nie Tan (8) and Wu Mendel method (10). The acquired data confirm the results obtained for the oscillating object under study.

CONCLUSIONS

The main goal of the article was to present a new Matlab toolbox to facilitate usage of a novel control method - type-2 fuzzy controller. In the first section the details of the created software were presented. Noteworthy is the fact that, unlike in other available toolboxes, it is easy to create and immediately use all the important parts a type-2 fuzzy controller. For example, a user can create custom membership functions for both inputs and outputs. The toolbox is designed in such a way that the obtained results of the simulation work can be easily transferred to the real world by generating controller code for rapid implementation by the PLC. Moreover, one of the most difficult operations in type-2 fuzzy logic systems is type reductionn. The 16 most popular methods have been created in the software, it it is also possible to modify them and create new ones.

The second part of the article presents the time performance results for the mentioned 16 reduction methods. The Nie-Tan method was the fastest one. In further research, the operation of a type-2 fuzzy controller was simulated using all available type-reduction methods. The step response analyze for an oscillating object model was used to compare the performance of the created controller. The Wu-Mendel and Nie-Tan method turned out to be the most effective for the researched oscillating object. Taking into consideration timing results and control quality the Nie-Tan method was distinguished. Having a new well-design toolbox allows in further research focus on modification of this method to achieve even better controller performance.

REFERENCES

1. Seising R., Lotfi Zadeh: fuzzy sets and systems. InComputer Aided Systems Theory–EUROCAST 2019: 17th International Conference, Las Palmas de Gran Canaria, Spain, February 17–22, 2019, Springer International Publishing.
2. Kaur J., Khehra B.S., Singh A. Significance of Fuzzy Logic in the Medical Science. In: Proc. of Computer Vision and Robotics: Proceedings of CVR 2021, Singapore: Springer Singapore, 2022 Mar 15, 497-509.
3. Sridharan M. Short review on various applications of fuzzy logic-based expert systems in the field of solar energy. International Journal of Ambient Energy. 2022; 43(1): 5112-28.
4. Bhuvaneswari S., Soujanya K.L. State of Art in Fuzzy Logic. Journal of Survey in Fisheries Sciences. 2023; 10(2S): 3562-82.
5. De A.K., Chakraborty D., Biswas A. Literature review on type-2 fuzzy set theory. Soft Computing. 2022; 26(18): 9049-68.
6. Karnik N.N., Mendel J.M. Introduction to type-2 fuzzy logic systems. In: Proc. of 1998 IEEE international conference on fuzzy systems proceedings. IEEE world congress on computational intelligence (Cat. No. 98CH36228) 1998 May 4, 2, 915-920, IEEE.
7. Karnik N.N., Mendel J.M. Operations on type-2 fuzzy sets. Fuzzy sets and systems. 2001 Sep 1; 122(2): 327-48.
8. Mendel J.M. Type-2 fuzzy sets: some questions and answers. IEEE Connections, Newsletter of the IEEE Neural Networks Society. 2003, 1: 10-3.
9. Wu D., Mendel J.M. Uncertainty measures for interval type-2 fuzzy sets. Information sciences. 2007 Dec 1; 177(23): 5378-93.
10. Changdar C., Mondal M., Giri P.K., Nandi U., Pal R.K. A two-phase ant colony optimization based approach for single depot multiple travelling salesman problem in Type-2 fuzzy environment. Artificial Intelligence Review. 2023; 56(2): 965-93.
11. Castro J.R., Castillo O., Martinez L.G. Interval Type-2 Fuzzy Logic Toolbox. Eng. Lett.. 2007; 15(1): 89-98.
12. Taskin A., Kumbasar T. An open source Matlab/Simulink toolbox for interval type-2 fuzzy logic systems. In: Proc. of 2015 IEEE Symposium Series on Computational Intelligence, 2015, 1561-1568.

13. D'Alterio P., Garibaldi J.M., John R.I., Wagner C. Juzzy constrained: Software for constrained interval type-2 fuzzy sets and systems in Java. In: Proc. of 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) 2020 Jul 19 (pp. 1-8).
14. Wagner C. Juzzy-a java based toolkit for type-2 fuzzy logic. In: Proc. of 2013 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ,) 2013 Apr 16, 45-52.
15. Wagner C., Miller S., Garibaldi J.M. A fuzzy toolbox for the R programming language. In: Proc. of 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), 2011 Jun 27, 1185-1192.
16. Haghrah A.A., Ghaemi S. PyIT2FLS: A new python toolkit for interval type 2 fuzzy logic systems. arXiv preprint arXiv:1909.10051. 2019.
17. Taskin A., Kumbasar T. An open source Matlab/Simulink toolbox for interval type-2 fuzzy logic systems. In: Proc. of 2015 IEEE Symposium Series on Computational Intelligence, 2015, 1561-1568.
18. Wagner C. Juzzy-a java based toolkit for type-2 fuzzy logic. In: Proc. of 2013 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ), 2013, 45-52.
19. Torshizi A.D., Zarandi M.H., Zakeri H. On type-reduction of type-2 fuzzy sets: A review. Applied Soft Computing. 2015; 27: 614-27.
20. Yeh C.Y., Jeng W.H., Lee S.J. An enhanced type-reduction algorithm for type-2 fuzzy sets. IEEE Transactions on Fuzzy Systems. 2010; 19(2):227-40.
21. Nie M., Tan W.W. Towards an efficient type-reduction method for interval type-2 fuzzy logic systems. In: Proc. of 2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence), 2008, 1425-1432.
22. Chen Y. Study on centroid type-reduction of interval type-2 fuzzy logic systems based on noniterative algorithms. Complexity. 2019; 2019: 1-2.
23. Wu L., Qian F., Wang L., Ma X. An improved type-reduction algorithm for general type-2 fuzzy sets. Information Sciences. 2022; 593: 99-120.
24. Li C., Yi J., Zhao D. A novel type-reduction method for interval type-2 fuzzy logic systems. In: Proc. of 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery 2008 Oct 18, 1, 157-161.
25. Coupland S., John R. A fast geometric method for defuzzification of type-2 fuzzy sets. IEEE Transactions on Fuzzy Systems. 2008; 16(4): 929-41.