Development of a fuzzy rule base design algorithm using a genetic algorithm and an inverted pendulum

Nikita Marushchenko[1*] and *Vladimir* Stanovov[2]

[1]Institute of Physics, FRC KSC SB RAS, st. Akademgorodok, 50, Krasnoyarsk, Russia
[2]Reshetnev Siberian State University of Science and Technology, Krasnoyarsky Rabochy Av, 31, Krasnoyarsk, Russia

Abstract. This paper presents the development of a fuzzy rule base construction algorithm that leverages a genetic algorithm to optimize control parameters for complex dynamic systems. The algorithm was tested on an inverted pendulum model, a classical nonlinear control problem, to identify chromosome configurations capable of stabilizing the pendulum in an upright position. Through extensive experimentation, the program successfully discovered such chromosomes that stabilized the pendulum from a variety of initial positions, including inverted (top-down) states, by forming the needed fuzzy rules base and fuzzy input and output terms. The results demonstrate the efficiency of the proposed approach in automating the design of fuzzy logic controllers using evolutionary methods.

1 Introduction

With the rapid development of computational intelligence technologies in recent decades, machine learning methods have found widespread application across a broad range of scientific and practical domains. These technologies, in particular, significantly accelerate problem-solving and improve accuracy compared to traditional methods, which are done manually. By automating processes, machine learning enables the efficient, rapid handling of large datasets, conserving both human and material resources. Furthermore, these technologies may identify new solutions for problems once considered complex or unsolvable, thus expanding their scope across various fields of science and practice.

One area where machine learning technologies are particularly beneficial is automatic control theory. This field often involves “black box” problems, which require developing a control system for a given structure without a mathematical model of internal processes, relying solely on input and output data. In this context, machine learning technologies facilitate a fundamental shift in the approach to solving such problems, potentially simplifying—or even fully automating—the process of constructing control systems for "black box" scenarios.

This paper introduces an algorithm that, without analyzing the system's model, generates a fuzzy rule base and fuzzy input and output terms to control a chosen physical model. Python

* Corresponding author: jeetee24@gmail.com

was used for support and utility tasks, such as rendering graphs and verifying results, while C++ served as the primary programming language for this work.

The selected physical model is the widely studied inverted pendulum with a rotating base - a nonlinear system commonly used in stabilization and control tasks, which requires complex algorithms and methods for effective stabilization [1]. The relevance of this work lies in its proposed method's ability to create control systems without specialized domain knowledge. This work builds upon the results presented in [2], creating a more sophisticated system with reduced human involvement. This is accomplished by having both fuzzy rules and fuzzy terms found automatically, without any prepared data.

1.1 Related work

The inverted pendulum is a nonlinear system, which makes its control challenging. Managing such systems requires the application of complex algorithms and methods [3]. Many real-world systems exhibit similar dynamics and stabilization requirements. For instance, controlling space rockets, robots, walking machines, and other engineering systems present challenges similar to those encountered with the inverted pendulum [4]. If an algorithm can effectively stabilize an inverted pendulum, there is a strong likelihood it can be adapted for other complex systems. The objective is to keep the pendulum vertical by rotating the horizontal base to which it is connected via a bearing or another element that allows free rotation. A simplified example of such a model is shown in Fig. 1.

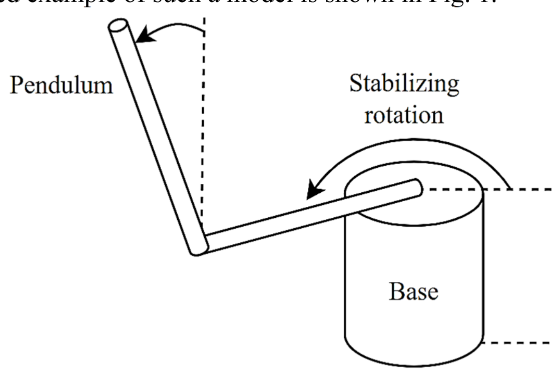

Fig. 1. Inverted pendulum model.

The next essential components for completing the entire algorithm are a fuzzy logic module and a genetic algorithm module. Fuzzy logic-based control mechanisms are well known for their flexibility, relatively clear structure, and high efficiency in various control problems [5, 6]. However, designing such systems typically requires expert knowledge to determine linguistic variables, fuzzy term granulation, and rule base design. One way to overcome this limitation is by using evolutionary algorithms, which solve complex optimization problems with algorithmically defined goal functions [7]. In this study, a genetic algorithm (GA) is applied to the automatic design of a Mamdani-type fuzzy controller [8], with the objective of finding such rule base configuration, which allows for the model to achieve its target condition.

1.2 Decoding chromosome into Fuzzy Rules

To connect the fuzzy logic and genetic algorithm modules - specifically, to enable the chromosomes of the genetic algorithm to interface with the fuzzy control module - it is necessary to convert the chromosome data into standard decimal numbers. Chromosomes, composed of simple sets of genes or bits (specifically, the digits "1" and "0"), must be transformed in a precise manner into sets of fuzzy terms and rules. The terms used in the fuzzy control module are defined as coordinate points in space. To encode data for three points and determine which output term corresponds to a given input term or terms, it is essential to develop an algorithm that enables these points to be represented within the chromosome.

By using an approach similar to converting binary representations to integer values, with the addition of modifiers (coefficients) and shifts, the required decoding ranges can be established. In this specific case, a tailored solution was designed, in which the chromosomes contain three sequential data fields where the input terms, output terms, and configurations are defined. The principle is illustrated in the Figure 2.

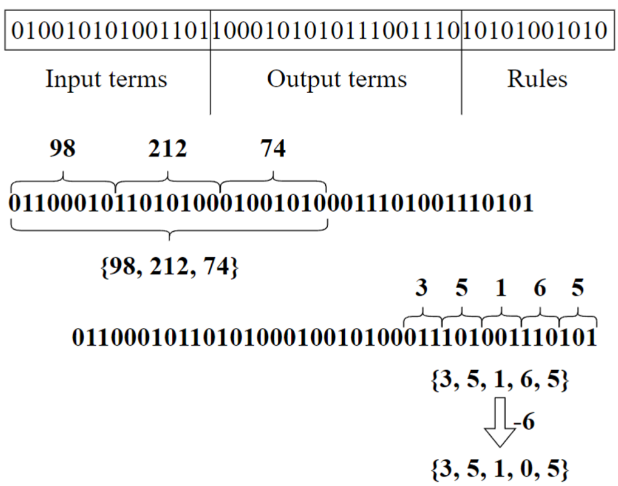

Fig. 2. Decoding Chromosome into Fuzzy parameters.

In Fig. 2, an example chromosome is shown, divided into three ranges. Each range contains a specific set of data that can be decoded into triplets, representing term points for the fuzzy control system. Additionally, at the end of the chromosome, a set of decoded numbers connects input and output fuzzy terms to form the rule database. Solutions, in some ways similar to this can be found in works [9, 10].

The flowchart of the complete algorithm is presented on the Figure 3.

The algorithm begins by generating an initial population and then determines the fitness of each individual. To do this, it decodes the chromosome to obtain parameters for the fuzzy control module, which are then used to stabilize the model. The quality of stabilization is assessed based on the characteristics of its oscillatory behavior, specifically the pendulum's angular deviation from the vertical position. After calculating fitness values for the entire generation, the algorithm performs tournament selection with two participants, creates offspring by applying one-point crossover, and applies mutation with the probability of 1/L, where L is a chromosome length. This process is repeated for a specified number of iterations.

1.3 Fitness function

A specific algorithm for determining the fitness of each individual was established, based on the number of pendulum rotations, the number of base rotations, the squared angular error values, and an increasing error weight with each iteration. This setup ensures that

stabilization processes have a diminishing impact on error assessment over time. Furthermore, each of these components in the calculation has its own weight.
The fitness function equation is presented below.

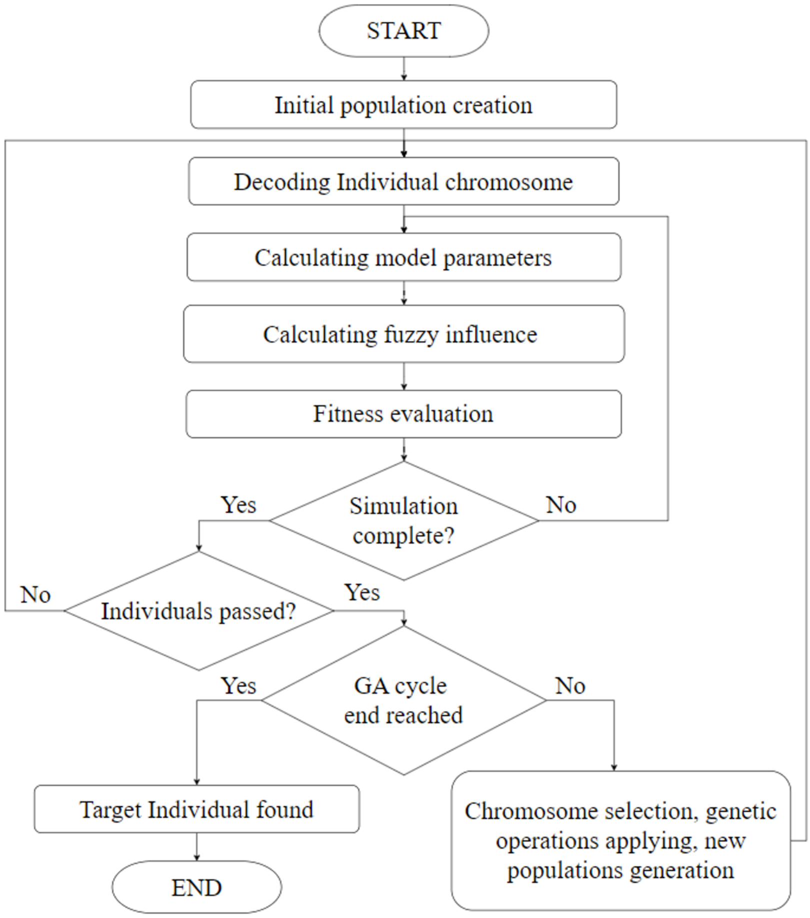

Fig. 3. Simplified program flowchart.

$$f_n = (x1_n * x1_m + x2_n * x2_m + \sqrt{x3_n * x3_n} * x3_m) * i / i_m \tag{1}$$

Formula's parameters $x1_n$, $x2_n$ and $x3_n$ are Base angle, Pendulum angle and Base speed of iteration i. $x1_m$, $x2_m$, $x3_m$ and i_m are weight coefficients for $x1_n$, $x2_n$, $x3_n$ and i parameters.

2 Experiments

After finalizing the algorithm, experiments were conducted using various coefficients for the weights of the fitness function components and different meta-parameters. Three groups of stabilization models emerged from the results: those that stabilized only the pendulum, those that stabilized only the base, and those in which both the base and pendulum were stabilized. This variability is linked to the differing weights in the fitness determination function. For effective stabilization, the system needs to balance both the base and the pendulum; therefore, the group where both the pendulum and base are stabilized is the most suitable for addressing the problem. The results of a stabilization experiment are shown in Figure 4.

 https://doi.org/10.1051/itmconf/20257204002

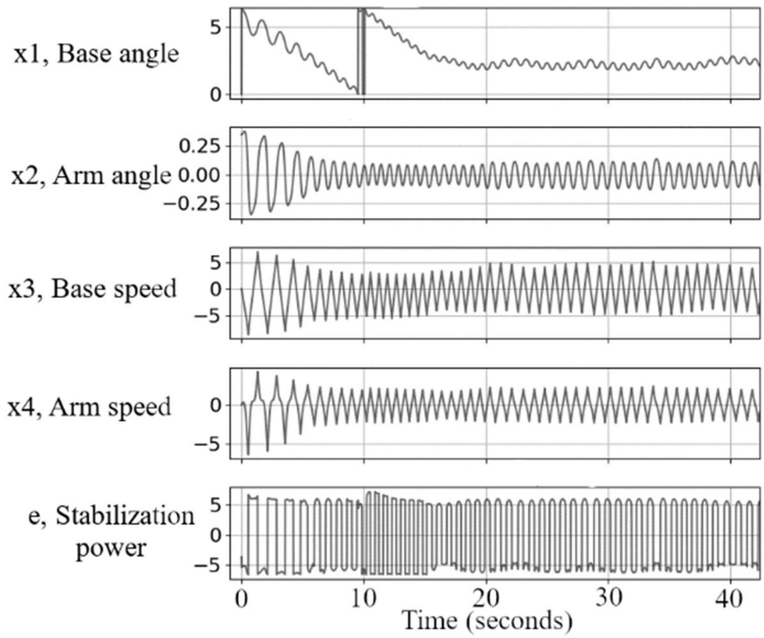

Fig. 4. Stabilizing the base and pendulum arm in vertical position.

The fig. 4 illustrates that the system stabilizes both the base—shown in the upper graph—and the pendulum. It is evident that the amplitude of oscillations decreases on both graphs and does not increase over time, while the pendulum's angle remains close to the desired value.

The system does not achieve perfect stabilization, as it is extremely sensitive near the angle values of 0 or 2π. This sensitivity arises because these points represent the vertical upright position, and the system frequently passes through them. Some of the boundaries of the input terms may be defined just before or after these points, but not precisely at them. Naturally, there may be individuals whose extreme term points correspond to these boundaries. By increasing the number of individuals and generations, it may be possible to find a solution with improved fitness.

All previous experiments assessed the feasibility of stabilizing the model with deviation angles up to 0.35π. Therefore, subsequent experiments were conducted to search for solutions that would enable the stabilization of the system from larger pendulum angles, including an angle of π, which represents the lowest vertical position of the pendulum to which it tends by default. Such a solution was found, and its characteristics are presented in fig. 5.

Fig. 5 presents experiments on stabilizing the pendulum from the downward position. Following the initiation of the stabilization mechanism, the amplitude of oscillations for both the base and the pendulum begins to increase, as indicated by the rising parameters. The graph of the pendulum's rotation angle shows that it swings with an increasing maximum angle until it reaches the upper point, where the amplitude of its oscillations decreases and continues to oscillate within a specific range around the upper point. The method proposed in this study identified rules and terms that allowed the pendulum to swing and flip vertically by rotating the base.

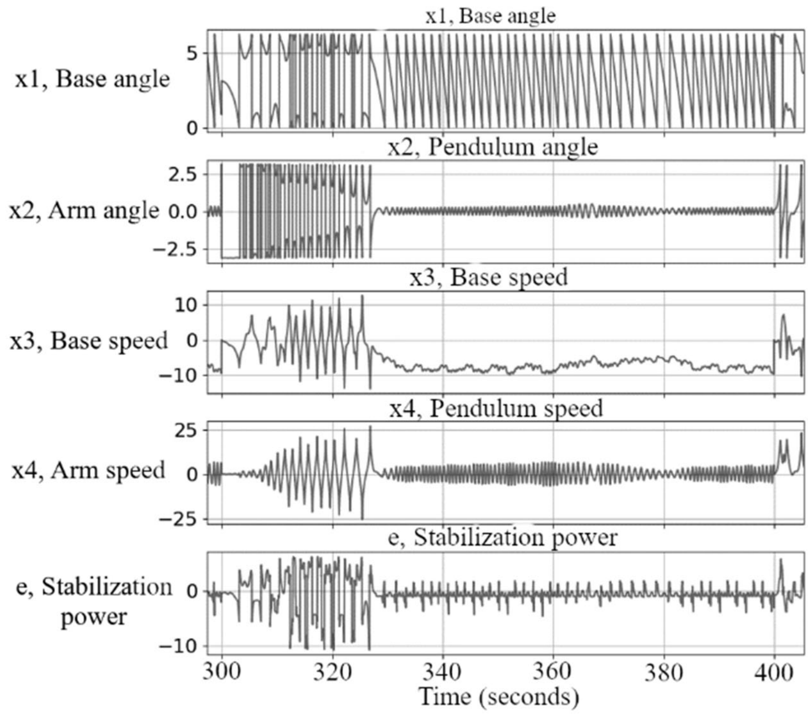

Fig. 5. Stabilizing pendulum by swinging from the rest position.

However, it is apparent that while the pendulum stabilizes, the base rotates around its axis without achieving stabilization, which contrasts with the experiments aimed at finding a model that stabilizes both elements. This issue likely arises from the insufficiency of output terms for precise balancing. Therefore, by increasing the hyperparameters of the fuzzy module, specifically the number of input and output terms, it may be possible to achieve stabilization of both the pendulum and the base from a broader range of initial angles. This was achieved when transitioning from a single input with six rules to a model with two inputs, each with six terms, and 36 rules.

Thus, while the operational range of the model has increased, the number of terms required for accurate balancing is still insufficient across the entire operational range of the model. All the aforementioned experiments stabilized the pendulum in the vertical upward position. However, the desired position depends solely on the fitness function employed. If necessary, stabilization could be achieved not relative to a specific point, but rather by seeking a solution that maintains a specific oscillatory process. Experiments to find such solutions were conducted, and successful models were found. Throughout experiments, population number used for genetic algorithm was in range from 500 to 1500, and generations number was from 500 to 1000.

3 Fuzzy rule base output

After obtaining a fuzzy rule base, its characteristics can be visualized as shown on fig. 6 and Table 1. These parameters represent a found model, which was stabilizing the arm vertically from different arm’s starting angles. The Figure 6 represents areas of input terms A and B (for case when input parameters were using values of both Base and Pendulum angles) and on the bottom graph Output terms can be seen. It needs to be mentioned, that while on Output terms e is defined between -0.75 and 1, applied value multiplied by 100 in calculations. Figure 7 represents a table, which shows corresponding Input and Output terms, forming the fuzzy rule base.

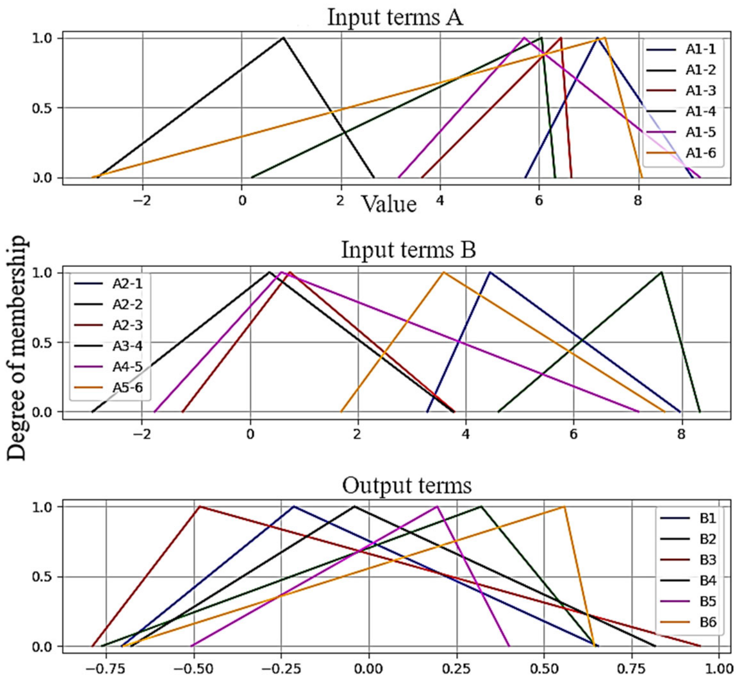

Fig. 6. Input and output terms founded by algorithm

It can be seen on Input terms A and B that the system finds areas near 0 or 2π, which was the target stabilization area. On output terms it is shown, that the system makes use of the almost whole range of possible stabilisation power e.

Table 1 shows which combination of input terms A and B (case A) results in which output rule (case B), thus making a rule base with the size of N rules, which comes from 6 by 6 unique input combinations. For example, this means that in case of rule 10 (N = 10) membership values for 1st input term 2 and 2nd input term 4 use output term 2.

Table 1. Fuzzy rules base founded by algorithm

N.	Input	Output	N.	Input	Output	N.	Input	Output
1	1 & 1	3	13	3 & 1	3	25	5 & 1	6
2	1 & 2	6	14	3 & 2	4	26	5 & 2	2
3	1 & 3	1	15	3 & 3	1	27	5 & 3	1
4	1 & 4	2	16	3 & 4	6	28	5 & 4	1
5	1 & 5	2	17	3 & 5	6	29	5 & 5	3
6	1 & 6	1	18	3 & 6	3	30	5 & 6	5
7	2 & 1	2	19	4 & 1	4	31	6 & 1	1
8	2 & 2	3	20	4 & 2	5	32	6 & 2	2
9	2 & 3	6	21	4 & 3	3	33	6 & 3	3
10	2 & 4	2	22	4 & 4	5	34	6 & 4	1
11	2 & 5	4	23	4 & 5	2	35	6 & 5	6
12	2 & 6	2	24	4 & 6	3	36	6 & 6	6

4 CONCLUSION

This study investigates the development of an algorithm for constructing bases of fuzzy rules using a genetic algorithm, exemplified by the solution to the problem of stabilizing an inverted pendulum. The principles of genetic algorithms and fuzzy logic were explored, and universal modules for fuzzy control and genetic algorithms were created, verifying their functionality. Experiments were conducted under various conditions and objectives, like stabilization of the model for different target conditions from different starting positions, leading to the design of bases of fuzzy rules with adjustments to the semantics of linguistic variables. An algorithm for decoding chromosomes into parameters of the fuzzy control module was developed. The novelty of this work lies in the ability to efficiently adjust the positions of fuzzy terms automatically. The resulting algorithm enables the creation of self-learning control algorithms for systems within a specific range of tasks, eliminating the need to manually construct the control system.

This work was supported by the Ministry of Science and Higher Education of the Russian Federation (grant No. 075-15-2022-1121).

REFERENCES

1. M.A. Fairus, Z. Mohamed, M.N. Ahmad, IOP Conf. Ser.: Mater. Sci. Eng. **53,** 012009 (2013)
2. V. Stanovov, S. Akhmedova, E. Semenkin, Automatic design of fuzzy controller for rotary inverted pendulum with success-history adaptive genetic algorithm in *Proceedings of the 2019 IEEE International Conference on Information Technologies* (Bulgaria, 2019)
3. Md. Akhtaruzzaman, A.A. Shafie, Modeling and control of a rotary inverted pendulum using various methods, comparative assessment and result analysis in *Proceedings of the 2010 IEEE International Conference on Mechatronics and Automation*, pp. 1342–1347 (2010)
4. L.B. Prasad, B. Tyagi, H.O. Gupta, Int. J. Autom. Comput. **11,** 661–670 (2014)
5. V. Kumar, A.P. Mittal, Int. J. Autom. Comput. **7,** 463–471 (2010)
6. K.M. Passino, S. Yurkovich, *Fuzzy Control* (Addison-Wesley Longman, Inc., California, 1998)
7. U. Bodenhofer, F. Herrera, Ten Lectures on Genetic Fuzzy Systems, Pre-prints of the International Summer School: Advanced Control – Fuzzy, Neural, Genetic, Slovak Technical University, Bratislava (1997)
8. E.H. Mamdani, S. Assilian, Int. J. Man-Machine Stud. **7,** 1–13 (1975)
9. V.V. Kureychik, S.I. Rodzin, Izv. Yuzhno-Fed. Univ. Tekhn. Nauki **7,** 13–21 (2010)
10. M. Oltean, Improving the Search by Encoding Multiple Solutions in a Chromosome 10.48550/arXiv.2110.11239