OPTIMIZED FUZZY RULE REDUCTION THROUGH KARNAUGH MAP-BASED SYSTEM

Mandy Woon Chian Wen[1], Azura Che Soh[2], Ribhan Zafira Abdul Rahman[3]

[1] *Student, Depart. of Electrical & Electronic Eng., Universiti Putra Malaysia, Selangor, Malaysia*
[2] *Lecturer, Depart. of Electrical & Electronic Eng., Universiti Putra Malaysia, Selangor, Malaysia*
[3] *Lecturer, Depart. of Electrical & Electronic Eng., Universiti Putra Malaysia, Selangor, Malaysia*

ABSTRACT

In recent years, fuzzy logic systems have been employed in various applications with great success, controlling systems, machines, and consumer products. Although these systems are well-developed, issues persist regarding poor and unreliable performance quality. The number of design rules increases, resulting in a decrease in the accuracy of the controller, and the execution time of the system rises with the growing number of fuzzy rules. The number of fuzzy rules further increases when applied to complex systems involving more input and output parameters. Therefore, it is necessary to reduce the number of fuzzy rules to enhance system performance, especially in reducing computation time during system operation. Several advanced techniques, such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and others, are utilized to reduce the number of fuzzy rules. All these techniques are algorithms that require programming and cannot be used for manual design. Consequently, the Karnaugh Map (K-MAP) systematic approach technique is applied in this study to decrease the number of fuzzy rules for the fuzzy controller. The K-MAP is one of the methods used to simplify fuzzy logic rules, which can be applied or designed programmatically or manually. The K-MAP is much simpler and does not require knowledge of Boolean algebraic theorems. It involves a smaller number of steps compared to other solutions. This research involves two different case studies with varying numbers of input variables and one output variable. The performance of the selected case study is compared, and computation times have been improved.

Keyword: - *Karnaugh Map (K-MAP), Fuzzy Rules, Optimized Rules, Fuzzy Controller*

---

1. INTRODUCTION

As technology continues to advance, the popularity of fuzzy logic systems is increasing. The number and diversity of applications for fuzzy logic have grown significantly. Commercially, fuzzy logic has been employed with great success to control machines and consumer products. In certain applications, fuzzy logic systems are simple to design and can be understood and implemented by those unfamiliar with fuzzy logic control. While the control system may not be optimal, it can be deemed acceptable. Fuzzy logic controllers often yield superior solutions compared to conventional control techniques.

The fuzzy logic method is integrated with the rule-based system, forming a crucial component within the structure of a fuzzy logic controller. This integration of rule-based systems in artificial intelligence (AI) has brought us a step closer to achieving the vision of creating machines capable of reasoning and decision-making akin to humans. Utilizing a set of predefined rules to process information and provide solutions, these systems have become indispensable tools for addressing complex challenges across diverse fields [1]-[5].

The benefit of employing rule-based systems is their ability to furnish a transparent and interpretable framework for decision-making, making them easier to maintain and update in comparison to other AI models employing more complex algorithms. However, a potential drawback of rule-based systems is their limitation in handling situations beyond the scope of predefined rules. In instances where rules are incomplete or incorrect, the system may offer inaccurate or incomplete solutions.

A rule-based system is the simplest form of artificial intelligence commonly used in industry. It encodes human knowledge to facilitate system execution. Essentially, a rule-based system consists of a set of 'If-Then' statements that utilize rules as coded knowledge for system operation. Also known as expert systems, these systems incorporate the knowledge of human experts into a set of rules to solve problems. Expert systems demonstrate consistent performance when exposed to the same data. Rule-based systems are easy-to-understand models and are highly efficient when applied to problem-solving.

The number of fuzzy logic rules primarily depends on the quantity of inputs and linguistic variables assigned to each input. However, an increased number of inputs can impact computational memory and result in increased time requirements, making the system take longer to produce results and leading to more complex hardware. A general control algorithm for a loop controller should ideally have a small number of tuning parameters and short computation time, considering limited memory and slower processors. Nevertheless, general fuzzy logic control involves numerous tuning parameters in the membership functions and control rules. Consequently, it becomes challenging for the control inputs of general fuzzy logic control to be computed within a short period. The speed of fuzzy control remains a significant concern when implementing fuzzy controls on general-purpose processors.

Various algorithms, including GA [6]-[7] and PSO [8]-[10], are commonly employed to minimize fuzzy rules. However, these algorithms require programming skills for implementation. The introduction of K-MAP provides an alternative for rule reduction, eliminating the need for programming expertise. Utilizing K-MAP as a systematic approach to reduce rules offers flexibility, allowing users to choose between programming implementation or manual rule reduction, depending on their preference and expertise.

This research aims to achieve the following objectives: primary, to minimize the rules of the fuzzy logic-based controller in the selected case study using a Karnaugh Map. Additionally, the research addresses the following sub-objectives: (i) To decrease the computation time of the fuzzy logic system and (ii) To design a fuzzy logic system with reduced rules without compromising its performance.

2. METHODOLOGY

The fuzzy logic system comprises four main components: Fuzzification, Knowledge Base, Inference Engine, and Defuzzification, each serving specific functions. Fuzzification is responsible for transforming crisp input numbers into fuzzy sets. Subsequently, the Knowledge Base initiates operations by storing IF-THEN rules provided, containing knowledge about the application domain and control goals. The Inference Engine executes inference for fuzzy control actions, comparing membership functions and generating qualified output memberships. Finally, the Defuzzification operation determines the crisp value from the output membership. The rule-based system in the inference process involves determining reduction rules by applying the K-MAP. The block diagram of the fuzzy logic system is depicted in Figure 1.

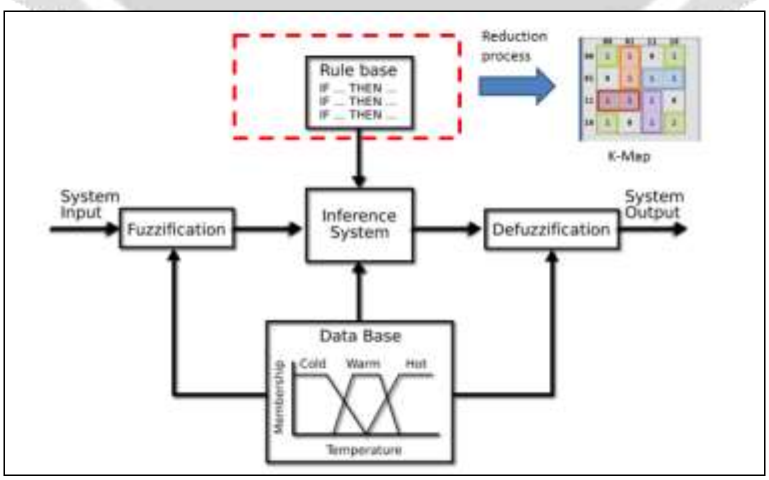

Figure -1: Block diagram of the fuzzy logic system

An illustrative application of the K-MAP technique involves a fuzzy system with two inputs, A and B, and one output, C. In Table 1, the input and output spaces are divided into two regions based on two linguistic values: Positive Small (PS) and Negative Small (NS).

Table -1: Value of each fuzzy set for input variables and output variable

Input/Output	**Variable**	**Fuzzy set**	**Range of values**
Input	A	Negative Small (NS)	0-5
		Positive Small (PS)	5-10
	B	Negative Small (NS)	0-5
		Positive Small (PS)	5-10
Output	C	Negative Small (NS)	0-5

The fuzzy IF-THEN rules are listed based on the input and output variables. To apply the K-MAP method for rule minimization in a fuzzy logic system, the variables of the fuzzy logic set are converted into binary values, either 0 or 1. Subsequently, the K-MAP reduces the rules in accordance with the IF-THEN rules.

The fuzzy rules are listed as follows:

1. If A is PS and B is NS, then C is PS;
2. If A is PS and B is PS, then C is PS;
3. If A is NS and B is PS, then C is NS;
4. If A is NS and B is NS, then C is NS.

The value of the variable can be represented as a binary number: NS is 1, and PS is 0. Figure 2 shows an example of the K-MAP.

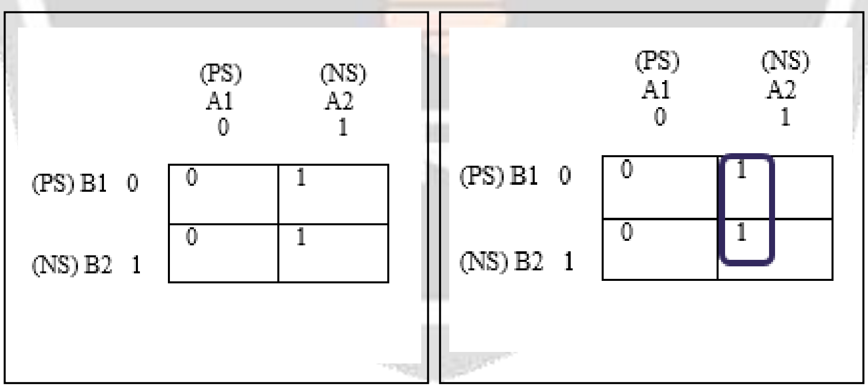

Figure -2: Example of the K-MAP

The reduced rules:

1. If A is NS and B is PS, then C is NS.
2. If A is NS and B is NS, then C is NS.

2.1 Case Study 1-Anti-Lock Braking System

The challenge with traditional braking systems is that the force applied by the brakes to the wheel must not surpass the frictional force between the wheel and the road. If the braking force exceeds the road's friction, the vehicle may start to slide, leading to accidents. Hence, an anti-lock braking system is crucial. The Anti-Lock Braking System has three inputs: distance between vehicles (in meters), applied brake force (in Newtons), and slip ratio. The output of the Anti-Lock Braking System is the speed of the vehicle. Fuzzy logic rules can be formulated based on these three inputs. Figure 3 illustrates the block diagram of the anti-lock braking system.

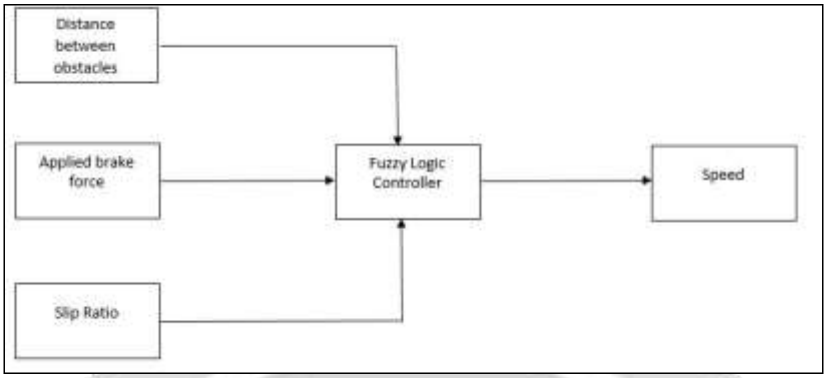

Figure -3: Block diagram of the anti-lock braking system

Each input space of the controller can be discretized into five fuzzy sets based on five linguistic values. The distance between obstacles is discretized as Extremely Near (EN), Very Near (VN), Near (N), Far (F), and Very Far (VF). However, the applied brake force and slip ratio are discretized into Very Low (VL), Low (L), High (H), Very High (VH), and Extremely High (EH). The output spaces of the controller can be discretized into six fuzzy sets with linguistic values defined as Zero (Z), Low (L), Medium (M), High (H), and Very High (VH). The range of values for input and output variables is tabulated in Table 2. With the information from Table 2, rules for the Anti-Lock Braking System can be formulated. The designed rules are listed in Table 3.

Table - 2: Values of each fuzzy set for input variables and the output variable for Case Study 1

Input/Output	Variable	Fuzzy set	Range of values
Input	Distance between obstacle (meters)	Extremely Near, EN	0 – 5
		Very Near, VN	6 – 10
		Near, N	11 – 20
		Far, F	21 – 30
		Very Far	31 – 40
	Applied brake force(N)	Very Low	0 – 5
		Low, L	6 – 10
		High, H	11 – 20
		Very High, VH	21 – 30
		Extremely High, EH	31 – 40
	Slip ratio	Very Low, VL	0 – 5
		Low, L	6 – 10
		High, H	11 – 20
		Very High, H	21 – 30
		Extremely High, EH	31 – 40
Output	speed (meter per seconds)	Zero, Z	0
		Low, L	1 – 10
		Medium, M	11 – 20
		High, H	21 – 30
		Very High, VH	31 – 40
		Extremely high, EH	41 – 50

Table - 3: Rules for the Anti-Lock Braking System

IF	Input 1 (Distance)	Condition	Input 2 (Force)	Condition	Input 3 (Ratio)	Condition	Output (Speed)
RULE 1	EN	AND	EH	AND	EH	THEN	Z
RULE 2	EN	AND	EH	AND	VH	THEN	Z
RULE 3	EN	AND	EH	AND	H	THEN	M
RULE 4	EN	AND	EH	AND	L	THEN	M
RULE 5	EN	AND	EH	AND	VL	THEN	H
RULE 6	EN	AND	VH	AND	EH	THEN	Z
RULE 7	EN	AND	VH	AND	VH	THEN	M
RULE 8	EN	AND	VH	AND	H	THEN	M
RULE 9	EN	AND	VH	AND	L	THEN	H
RULE 10	EN	AND	VH	AND	VL	THEN	H
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
RULE 120	VF	AND	L	AND	VL	THEN	EH
RULE 121	VF	AND	VL	AND	EH	THEN	H
RULE 122	VF	AND	VL	AND	VH	THEN	VH
RULE 123	VF	AND	VL	AND	H	THEN	EH
RULE 124	VF	AND	VL	AND	L	THEN	EH
RULE 125	VF	AND	VL	AND	VL	THEN	EH

An assumption is made that the values of input and output are crisp values without considering overlapping. Thus, each of the variables can be represented as binary numbers: 000 for Z, 001 for L, 010 for M, 011 for H, 100 for VH, and 101 for EH. Other binary numbers (110 and 111) that are not used are assigned as "don't care" values (X). The inputs and output are named with symbols to simplify calculations. The inputs, distance (D), force (F), and slip ratio (C), are represented by A, B, and C, respectively, while the output, speed, is represented by D.

As an example, the first rule designed is: "If the distance between obstacles is extremely near, applied brake force is extremely high, and the slip ratio is extremely high, then the speed is zero." The rule is represented as: "If A is 000, B is 000, and C is 000, then D is 000.

The verbal description of the Anti-Lock Braking System designed rules is translated into algebraic expressions and depicted in the K-MAP, as illustrated in Figure 4. The symbols for the variables are listed in Table 4.

Table - 4: Symbols of the variables for Case Study 1

Binary Numbers	Distance, D	Force, F	Slip ratio	Speed
000	A1	B1	C1	D1
001	A2	B2	C2	D2
010	A3	B3	C3	D3
011	A4	B4	C4	D4
100	A5	B5	C5	D5
101	A6	B6	C6	D6
110	X	X	X	X
111	X	X	X	X

The following equation is derived for the output membership function, D. The Boolean expression for speed can be expressed as:

$$D = \sum_{n=2}^{6} An \sum_{n=2}^{6} Bn \sum_{n=2}^{6} Cn \tag{1}$$

Where n represents numbers ranging from 2 to 6 for the inputs A, B, and C.

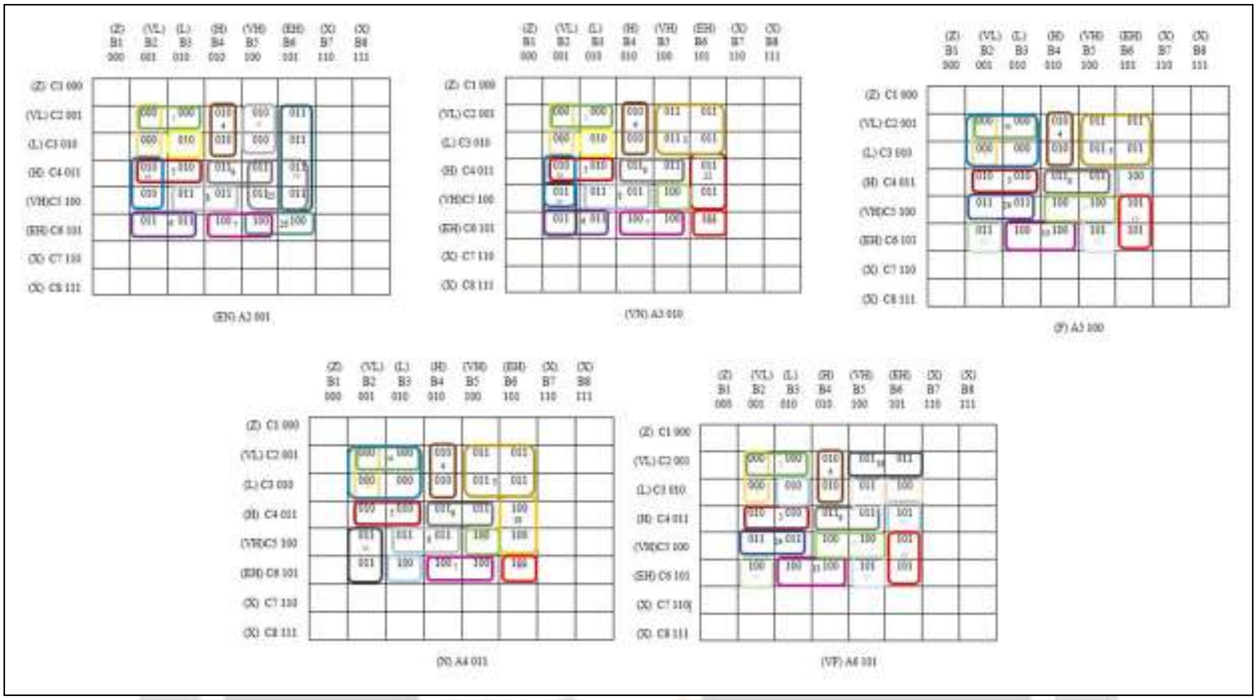

Figure - 4: Minimum map to obtain a standard Sum of Product(SOP) expression for Case Study 1.

As an example, the output for rule 1 is A2B2C2. The overall output membership function can be expressed as the product of the summation of An, Bn, and Cn and these outputs are then entered into the K-MAP. The K-MAP is designed for 5 variables, and blank cells indicate 'don't care' conditions. The groupings for the same output bit can be derived from the constructed K-MAP, resulting in the formation of the minimum map. The minimum map is depicted in Figure 4, where cells with the same binary values are combined to create the maximum groups. Each group includes the largest possible number of cells with the same binary values. The minimum product term for each group is then generated, forming the sum of product (SOP) expression. These SOP expressions are consolidated to derive a minimized Boolean expression for the speed output as shown below:-

D= B2 (C2+C3) + C2 (B2+B3) + (A4+A5) (B2B3C2C3)+B4(C2+C3) +C4(B2+B3) + A2B5(C2+C3) + B3C3(A2+A3+A6) +(A2B2)(C4+C5) +B5B6C2C3(A3+A4+A5)+(B2+B3)(A2C6+A3C6)+ (B3+B4)(A2C5+A3C5+A4C5) + C4(B4+B5) + A2B6(C2+C3+C4+C5) + A3B6(C4+C5) + (B2+B3)(A5C5+A6C5) + A4B2(C5+C6) + A6C2(B5+B6) + A2B5B6C4C5 + A3B2(C5+C6) + A5B2C6 + A6B5C3+(B4+B5)(A2C6+A3C6+A4C6) + (B3+B4)(A5C6+A6C6) +(B4+B5)(A5C5+A6C5) + A4B6(C4+C5) + (B5C5)(A3+A4) + A2C6(B5+B6) + A4B3C6 + A5B6C4 + A6B6C3 + A6B2C6+ (C5+C6)(A5B6+A6B6) + (B6C6)(A3+A4) + (B5C6)(A5+A6) + A6B6C4 (2)

The fully specified rules mentioned above can be translated from algebraic expressions to a verbal description of the anti-lock braking system. These rules are listed in Table 5. The designed rules for the Anti-Lock Braking System have been reduced from 125 to 89, resulting in a reduction of 36 rules, representing a 28.8% decrease from the original set of rules.

Table -5: Optimized rules for the anti-lock braking system

IF	Input 1 (Distance)	Condition	Input 2 (Force)	Condition	Input 3 (Ratio)	Condition	Output (Speed)
RULE 1	X	AND	EH	AND	EH	THEN	Z
RULE 2	X	AND	EH	AND	VH	THEN	Z
RULE 3	X	AND	EH	AND	H	THEN	M
RULE 4	X	AND	VH	AND	EH	THEN	Z
RULE 5	X	AND	VH	AND	H	THEN	M
RULE 6	X	AND	H	AND	EH	THEN	M
RULE 7	X	AND	H	AND	VH	THEN	M
RULE 8	X	AND	H	AND	H	THEN	H
RULE 9	X	AND	L	AND	H	THEN	H
RULE 10	EN	AND	EH	AND	L	THEN	M
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
RULE 84	VF	AND	L	AND	VL	THEN	EH
RULE 85	VF	AND	VL	AND	EH	THEN	H
RULE 86	VF	AND	VL	AND	VH	THEN	VH
RULE 87	VF	AND	VL	AND	H	THEN	EH
RULE 88	VF	AND	VL	AND	L	THEN	EH
RULE 89	VF	AND	VL	AND	VL	THEN	EH

2.2 Case Study 2- Load Elevator Control System

There is a consistent rise in energy consumption, driven by the growing demand to fulfil various comfort requirements in the construction of an increasing number of buildings, addressing the fundamental need for accommodation. The uninformed utilization of energy resources and the natural environment to enhance living standards in buildings disrupts the world's natural balance irreversibly. Therefore, a system for controlling load elevators has been designed.

The controlling of load elevators system features two inputs: Building Height (m) and Load Weight (kg). The system's output is the Power (kW) of the motor. Fuzzy logic rules can be formulated based on these inputs. Figure 5 illustrates the block diagram of the controlling of load elevators, while Table 6 provides the values for each fuzzy set corresponding to the input variables and the output variable for the Controlling of Load Elevators System. Additionally, Table 7 outlines the designed rules for controlling load elevators.

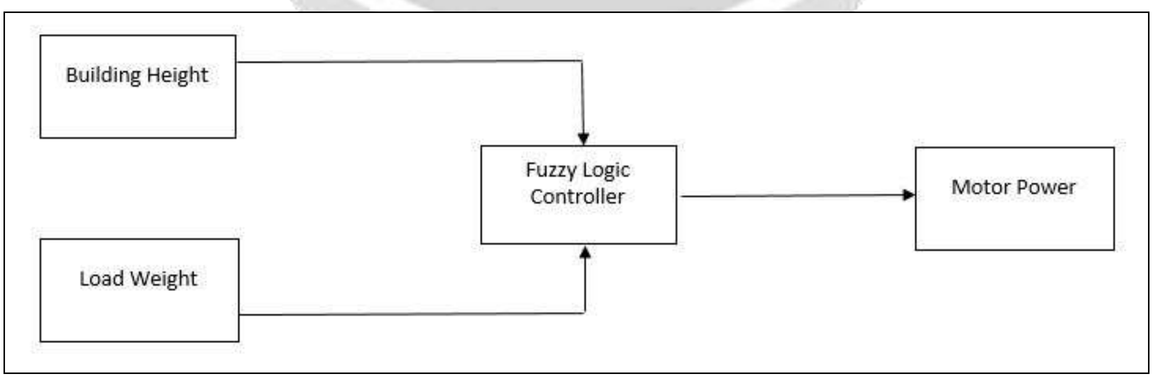

Figure - 5: Block diagram of controlling of load elevators

Table - 6: Values of each fuzzy set for input variables and the output variable for Case Study 2

Input/Output	Variable	Fuzzy set	Range of values
Input	Building Height (m)	Very Low, VL	0-20
		Low, L	21-40
		Middle, M	41-60
		High, H	61-80
		Very High, VH	81-100
	Load Weight (kg)	Very Light, VL	0-100
		Light, L	101-200
		Middle, M	201-300
		Heavy, H	301-400
		Very Heavy, VH	401-500
Output	Motor Power (kW)	Very Few Powerful, VFP	0-10
		Few Powerful, FP	11-20
		Middle Powerful, MP	21-30
		Powerful, P	31-40
		Very Powerful, VP	41-50
		Very Very Powerful, VVP	51-60

Table - 7: Rules for the load elevator control system

IF	Input 1 (Height)	Condition	Input 2 (Weight)	Condition	Output (Power)
RULE 1	VL	AND	VL	THEN	VFP
RULE 2	VL	AND	L	THEN	VFP
RULE 3	VL	AND	M	THEN	MP
RULE 4	VL	AND	H	THEN	P
RULE 5	VL	AND	VH	THEN	VVP
RULE 6	L	AND	VL	THEN	VFP
RULE 7	L	AND	L	THEN	MP
RULE 8	L	AND	M	THEN	MP
RULE 9	L	AND	H	THEN	P
RULE 10	L	AND	VH	THEN	VVP
RULE 11	M	AND	VL	THEN	MP
RULE 12	M	AND	L	THEN	MP
RULE 13	M	AND	M	THEN	P
RULE 14	M	AND	H	THEN	VP
RULE 15	M	AND	VH	THEN	VVP
RULE 16	H	AND	VL	THEN	P
RULE 17	H	AND	L	THEN	P
RULE 18	H	AND	M	THEN	P
RULE 19	H	AND	H	THEN	VP
RULE 20	H	AND	VH	THEN	VVP
RULE 21	VH	AND	VL	THEN	P
RULE 22	VH	AND	L	THEN	VP
RULE 23	VH	AND	M	THEN	VVP
RULE 24	VH	AND	H	THEN	VVP
RULE 25	VH	AND	VH	THEN	VVP

An assumption is made that the values of input and output are crisp, without considering overlapping. Thus, each variable can be represented as binary numbers: 000 for VFP, 001 for FP, 010 for MP, 011 for P, 101 for VP, and 110 for VVP. The table displays the symbols of variables, and Table 8 illustrates the symbols for controlling elevator loads. Binary numbers 111 and 100, which are not utilized, are designated as don't care values (X). Inputs and outputs are denoted by symbols for simplifying calculations. The inputs, building height (H) and load weight (W), are represented by A and B, while the output, motor power (P), is denoted by C. For instance, the initial rule crafted is: "If the building height is very low and the load weight is very light, then the motor power is very weak." This rule is expressed as: "If A is 000 and B is 000, then C is 000."

Table - 8: Symbols of the variables for Case Study 2

Binary Numbers	Building Height, H	Load Weight, W	Motor Power, P
000	A1	B1	C1
001	A2	B2	C2
010	A3	B3	C3
011	A4	B4	C4
100	A5	B5	C5
101	A6	B6	C6
110	X	X	X
111	X	X	X

The following equation defines the output membership function, C. The Boolean expression for motor power can be represented as:

$$C = \sum_{n=2}^{6} An \sum_{n=2}^{6} Bn \tag{3}$$

where, n represents values ranging from 2 to 6 for the inputs A and B.

As an illustration, the output for Rule 1 is represented as A2B2. The overall output membership function is expressed as the product of the summation of An and Bn, and these results are input into a 5-variable Karnaugh Map (K-MAP). The K-MAP, which includes blank cells to indicate 'don't care' conditions, facilitates the grouping of cells with the same output bit. By analyzing the constructed K-MAP, groupings for cells with the same binary values are identified to obtain the minimum map. Figure 6 illustrates the minimum map used to derive a standard Sum of Products (SOP) expression for Case Study 2. Cells with identical binary values are combined to form maximum groups, incorporating the largest possible number of identical binary values.

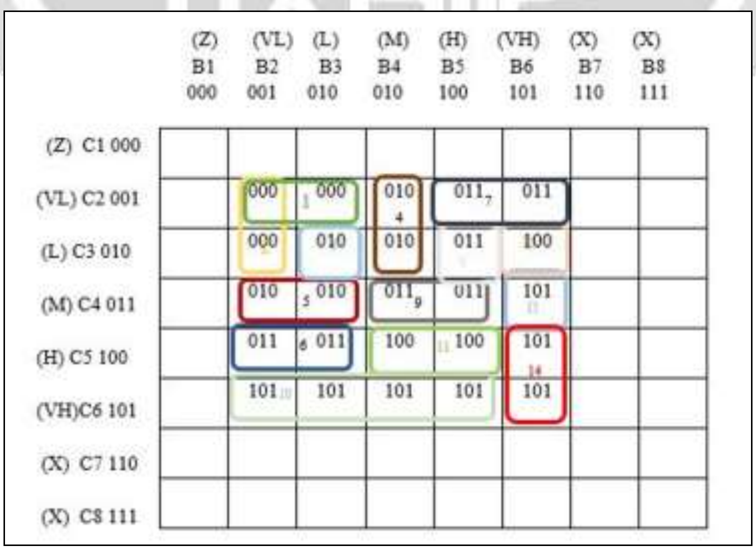

Figure- 6: Minimum map to obtain a standard SOP expression for Case Study 2

The minimum product term for each group is generated, resulting in the formation of the SOP expression as shown below:

$$C= B2\ (A2+A3) + A2\ (B2+B3) + A3B3 + A4\ (B2+B3) + B4\ (A2+A3) + B5\ (A2+A3) + B2\ (A5+A6) + A5B3 + B4\ (A4+A5) + B5\ (A4+A5) + A6B3 + B6\ (A2+A3+A4+A5) + A6B4 + A6\ (B5+B6) \quad (4)$$

The fully specified rules outlined above can be translated from the algebraic expression into a verbal description of the designed rules for the controlling of load elevators system. These rules are organized in Table 9. The rules for the controlling of load elevators system have been streamlined from 25 to 21, resulting in a reduction of 4 rules, equating to a 16% decrease from the original set of rules.

Table - 9: Optimized rules for the load elevator control system

IF	Input 1 (Height)	Condition	Input 2 (Weight)	Condition	Output (Power)
RULE 1	VL	AND	VL	THEN	VFP
RULE 2	VL	AND	L	THEN	VFP
RULE 3	L	AND	VL	THEN	VFP
RULE 4	L	AND	L	THEN	MP
RULE 5	M	AND	VL	THEN	MP
RULE 6	M	AND	L	THEN	MP
RULE 7	VL	AND	M	THEN	MP
RULE 8	L	AND	M	THEN	MP
RULE 9	VL	AND	H	THEN	P
RULE 10	L	AND	H	THEN	P
RULE 11	H	AND	VL	THEN	P
RULE 12	VH	AND	VL	THEN	P
RULE 13	H	AND	L	THEN	P
RULE 14	M	AND	M	THEN	P
RULE 15	H	AND	M	THEN	P
RULE 16	M	AND	H	THEN	VP
RULE 17	H	AND	H	THEN	VP
RULE 18	VH	AND	L	THEN	VP
RULE 19	X	AND	VH	THEN	VVP
RULE 20	VH	AND	M	THEN	VVP
RULE 21	VH	AND	H	THEN	VVP

3. RESULTS AND DISCUSSIONS

In the simulation results for Case Study 1, a focused effort was made to enhance the precision of the developed anti-lock braking system. The original rules and a reduced set of rules were applied and systematically compared. Figure 7 illustrates the 3D surface view of the anti-lock braking system, showcasing the original rules on the left and the reduced rules on the right. The figures clearly depict similar performance between the two rule systems. A more detailed comparison of output, as shown in Figure 8, reinforces the finding that there is no substantial difference between the original rules system and the reduced rules system. Moreover, the reduced rules exhibit comparable performance to the original rules, indicating that the system's overall performance remains unaffected. However, a quantitative analysis in Table 10, outlining the performance measures for Case Study 1, reveals that while the output variables remain consistent between the original and reduced rule systems, there is a notable reduction in computation time.

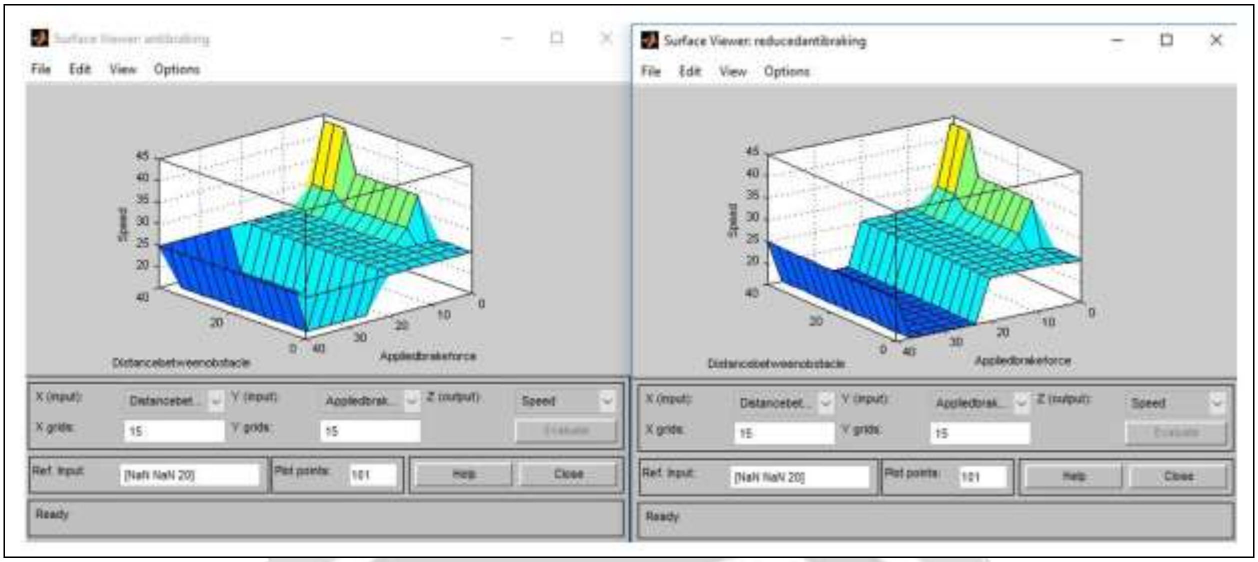

Figure-7: The 3D surface representation of the anti-lock braking system, depicting the original rules on the left and the reduced rules on the right.

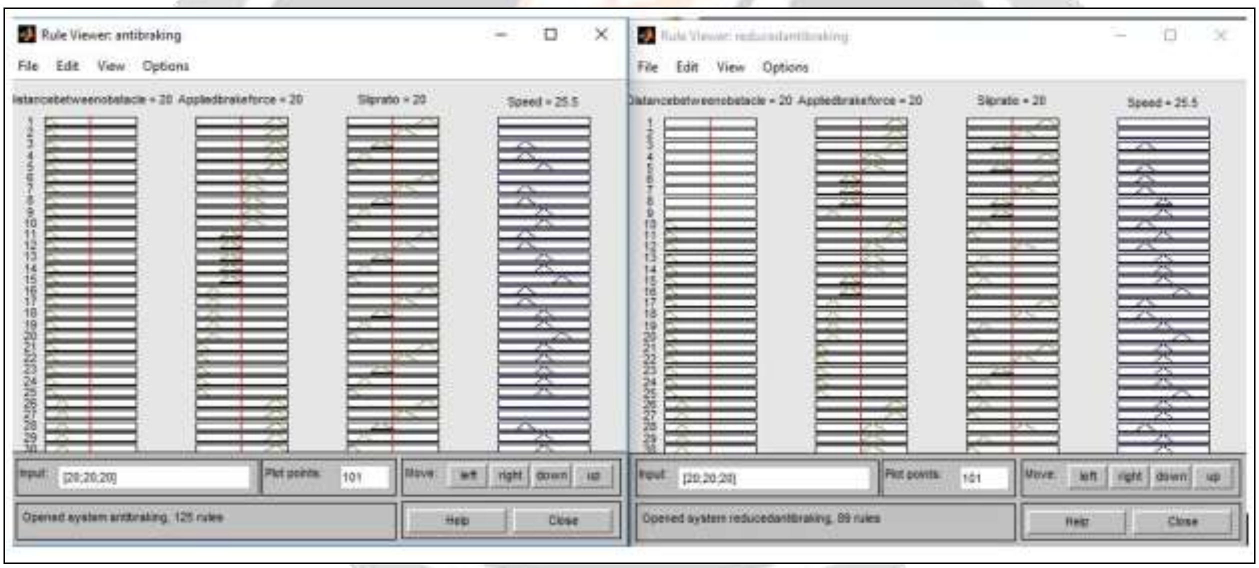

Figure-8: The original rules (left) and reduced rules (right) for the anti-lock braking system

Table-9: Performance measures for Case Study 1

	Variable	Original rules system (125 rules)	Reduced rules system (89 rules)
Input	Distance between obstacle(m)	20	20
	Applied brake force (N)	20	20
	Slip ratio	20	20
Output	Speed (meter per seconds)	25.5	25.5
Computation time		0.204319 s	0.159053 s

In Case Study 2, as depicted in Figures 9 and 10, the results indicate a negligible difference and tolerance between the original rules system and the reduced rules system. The reduced rules exhibit a performance comparable to the original rules, allowing the system to operate effectively with both the original and reduced rule sets for input and output variables.

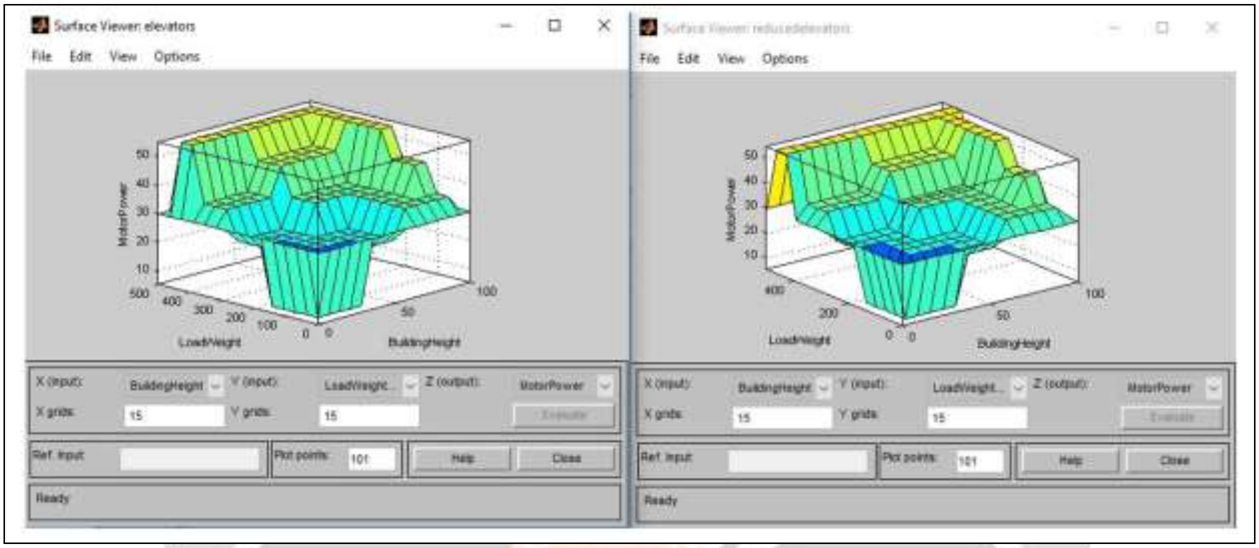

Figure-9: The 3D surface representation of the load elevator control system, depicting the original rules on the left and the reduced rules on the right

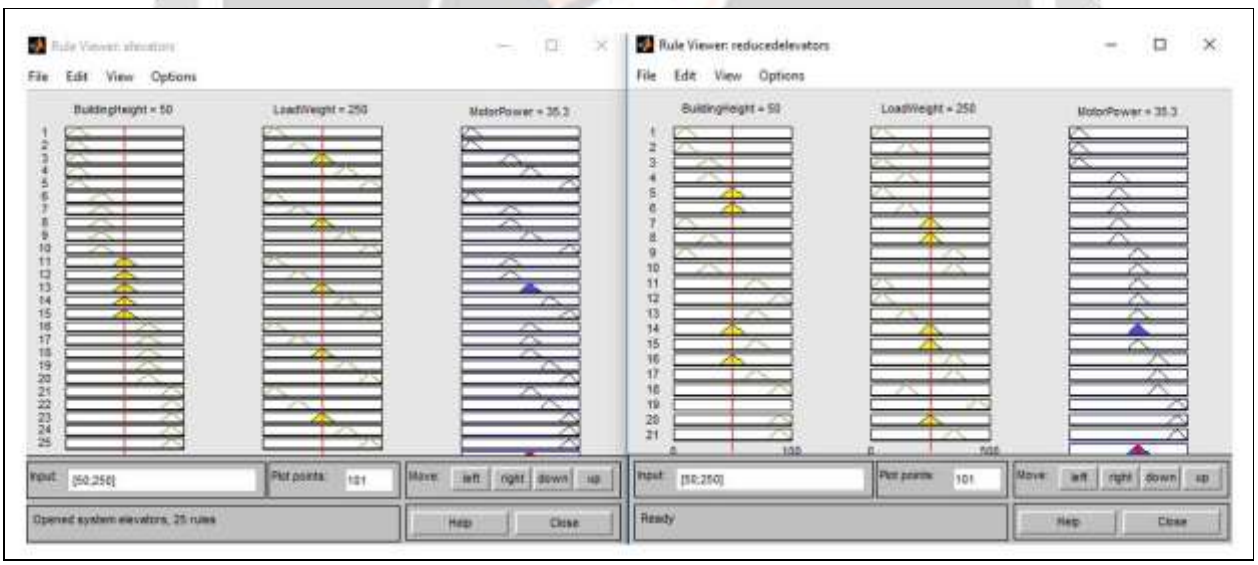

Figure-10: The original rules (left) and reduced rules (right) for the load elevator control system

Table 10 outlines the performance measures for the controlling of load elevators system. The output variables remain consistent between the original rules system and the reduced rule system, with no discernible difference or tolerance observed in the results. Consequently, the application of reduced rules not only reduces expensive computation time but also maintains similar performance to the original rules, ensuring the effective operation of the system. In conclusion, similar to Case Study 1, the system demonstrates enhanced computational efficiency with reduced rules, resulting in faster operation without compromising performance.

Table-9: Performance measures for Case Study 2

	Variable	Original rules system (25 rules)	Reduced rules system (21 rules)
Input	Building Height (m)	50	50
	Load Weight (kg)	250	250
Output	Motor Power (kW)	35.3	35.3
Computation times		0.126279 s	0.119308 s

4. CONCLUSIONS

The successful reduction of the number of selected case study fuzzy rules using K-Map led to an evaluation and comparison of the performance of the selected case study fuzzy-rules based controller with reduced rules against the controller with the original rules.

Simulation results indicate that the selected case study fuzzy-rules based controller with reduced rules exhibits similar performance to the original rules. This suggests that the reduction in rules does not adversely impact system performance. Furthermore, there is a decrease in the time required for computing executions. Consequently, this approach offers a cost-effective and efficient design for the fuzzy rule-based controller.

5. REFERENCES

[1] F. Liu, W. Sun, M. Heiner and D. Gilbert (2021), “Hybrid modelling of biological systems using fuzzy continuous Petri nets”, Briefings in Bioinformatics, Vol. 22, Issue 1, pp 438–450.

[2] A. Ren, J. Bai and J. Ma (2020), “Research on intelligent transportation system based on fuzzy neural network”, IOP Conf. Series: Materials Science and Engineering 768 (2020) 062114, pp 1-7.

[3] M. Hernández-Hernández, L. A. B. Cruz and L. Cobián-Romero (2023),”Improvement of validated manufacturing processes with fuzzy logic”, IntechOpen, DOI: 10.5772/intechopen.113302.

[4] A. Blasi (2018), “Scheduling food industry system using fuzzy logic”, Journal of Theoretical and Applied Information Technology, Vol.96. No 19, pp. 6463-6473.

[5] T. AL-Bazali and M. Al-Zuhair, (2022), “The use of fuzzy logic to assess sustainability of oil and gas resources (R/P): technical, economic and political perspectives”, International Journal of Energy Economics and Policy, 12(1), pp. 449-458.

[6] S.N. Veerababu, K. Roja, M.V.K. Reddy, K.M.K.Reddy and A. Ramesh (2020), “Optimization of fuzzy inference system using genetic algorithm”, Advanced in Intelligent and Computing (Book series), Vol.1248, Springer.

[7] M. T. Braz-César, P. L.P. Folhento and R. C. Barros (2018), “Fuzzy controller optimization using a genetic algorithm for non-collocated semi-active MR based control of a three-DOF framed structure”, 13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO) June 4-6, 2018, Ponta Delgada, Azores, Portugal.

[8] L. Mao, Q. Chen and J. Sun (2020), “Construction and optimization of fuzzy rule-based classifier with a swarm intelligent algorithm”, Hindawi Mathematical Problems in Engineering, Vol. 2020, pp. 1-18.

[9] D. S. Mai, T.H. Dang and L. T. Ngo (2021), “Optimization of interval type-2 fuzzy system using the PSO technique for predictive problems”, Journal of Information and Telecommunication, Vol. 5, Issue 2, pp. 197-213.

[10] T.A.Tran (2018), “The optimization of marine diesel engine rotational speed control process by fuzzy logic control based on particle swarm optimization algorithm”, Future Internet, 10(10), 99, pp 1-18.