Development of a Climate Equipment Parameter Acquisition System using PID and Fuzzy Logic Controllers to Improve Energy Efficiency

Marina Moseva
Department of Mathematical Cybernetics and Information Technologies, Moscow Technical University of Communications and Informatics, Russia
m.s.moseva@mtuci.ru (corresponding author)

Sergey Simonov
Department of Mathematical Cybernetics and Information Technologies, Moscow Technical University of Communications and Informatics, Russia
s.e.simonov@mtuci.ru

Mikhail Gorodnichev
Department of Mathematical Cybernetics and Information Technologies, Moscow Technical University of Communications and Informatics, Russia
m.g.gorodnichev@mtuci.ru

Received: 19 June 2024 | Revised: 31 July 2024 | Accepted: 3 August 2024

 DOI: https://doi.org/10.48084/etasr.8182

ABSTRACT

Today, energy-efficient resource management is an important task. This study aims to improve the energy efficiency of the cooling system of a technical room by developing a transparent and explainable temperature adaptation tuning algorithm based on the combination of PID control and fuzzy logic methods. This work focuses on the design and development of a hardware and software system consisting of a microcontroller and a temperature sensor. This paper analyzes temperature control based on PID and fuzzy controllers and proposes a combined method to allow for more accurate temperature control tuning. The experimental results show that the combined method reduces the rise time by at least 5%, the stabilization time by at least 17%, and the power consumption by at least 21%.

Keywords-PID controller; fuzzy logic; air conditioning system; data center; energy efficiency; thermal management

I. INTRODUCTION

As the computing power of Data Processing Centers (DPCs) develops, power consumption also increases. The distribution of the power consumption of a typical data center is as follows: approximately 60% for information technology equipment, approximately 35% for cooling, and approximately 15% for lighting and the needs of other power consumers. Cooling systems in data centers aim to maintain specific temperatures. A widely used solution in this area is the Proportional Integral Derivative (PID) controller, a device for automatically maintaining one or more parameters within a specified range. Current data centers use a variety of cooling systems, but fan coolers perform the mechanical movement of airflow and consume most of the power. Temperature management in data centers means effectively removing excess heat from the components as quickly and efficiently as possible. This study focuses on the cooling subsystem. The cooling system accounts for a large share of the energy consumption, namely 1/3 of the total energy consumption. For this reason, optimizing the processes of the cooling system is the key to improving the energy efficiency of the entire data center.

An effective approach to temperature management for data centers is based on the understanding that a solution does not necessarily fit all applications. Different systems use different methods to keep the temperature within the desired range, both at the hardware level of the climate system and at the software level. At the software level, the leader is the PID controller, an algorithm that automatically maintains one or more parameters within a specified interval. Typical modern temperature control uses the PID algorithm to provide accurate temperature control under varying environmental conditions [1-4]. Other algorithms can be equally successful, but the PID algorithm is the most widely used. Proper tuning of the temperature control algorithm requires some trade-off between fast response and precise control. The main parameter in the controller that affects the energy efficiency of the data center is the response time to changes in the external environment, i.e., temperature fluctuations. The response time of a PID controller depends on the selected coefficients, which are usually set manually. Using properly configured PID controllers shows good results when it comes to linear temperature control [5].

Predictive control based on machine learning techniques uses a detailed nonlinear dynamic model (including system boundaries, saturations, and hysteresis) of a pulsed power converter to predict its future behavior when each possible switching state or vector [6-9] is applied. For all possible switching states (vectors), the control errors are evaluated, and the vector that results in the minimum value of a suitable cost function is applied to the converter. In contrast, fuzzy control of switching converters does not require models, parameters, or operating conditions of the converter, but only expert knowledge of the converter dynamics [10-12]. Fuzzy controllers can be used in a variety of switching converters with little adaptation, as they are derived from the knowledge of the system dynamics using adaptive control. The resulting fuzzy control rules can be embedded in a decision reference table, where the control processor simply selects the control input corresponding to the selected measurements. Fuzzy controllers are practically robust to fluctuations in the system parameters since they do not consider their values. The use of fuzzy logic and PID controllers to adjust various parameters of complex systems shows good results in modeling HVAC systems [13], tuning PID controllers [14], controlling temperature in an incubator [15], and regulating water levels in a tank [16]. Some studies compared the performance of fuzzy PID controllers and machine learning algorithms to solve the problem of tuning system parameters, showing that fuzzy logic performed better [17, 18]. Based on the analysis of the applicability of different algorithms for solving the problem of parameter tuning to environmental conditions, it is proposed to consider the combination of algorithms that have shown themselves to be the best for solving the problem of temperature control.

This study proposes a transparent and explainable temperature adaptation tuning algorithm based on the combination of PID control and fuzzy logic methods. A prototype system was developed to investigate and develop a transparent algorithm for adaptive temperature control. The developed algorithm does not use machine learning algorithms, as they need a large amount of data to be trained to understand what to base the decision on, which prevents them from being used on devices with low processing power. Using a combination of PID controllers and fuzzy logic improved the system parameters, as the rise time was reduced by more than 5%, the stabilization time was reduced by more than 17%, and power consumption was reduced by more than 21%.

II. MATERIALS AND METHODS

The experimental system consists of hardware and software. Hardware includes an ESP32 microcontroller, a BME280 temperature and humidity sensor, a Snowman PWM cooler, an ESP32 development board (DevKit), and an ACS712 current sensor. The ESP32 microcontroller is a low-cost System-on-Chip (SoC) microcontroller from Espressif Systems [19]. It comes in both single-core and dual-core variants of Tensilica's 32-bit Xtensa LX6 microprocessor with built-in Wi-Fi and Bluetooth. Espressif Systems has released several ESP32-based modules, and one popular option is the ESP-WROOM-32, which consists of an ESP32 SoC, a 40 MHz crystal oscillator, a 4 MB Flash IC, and some passive components. The BME280 sensor module reads barometric pressure, temperature, and humidity [20]. Since pressure changes with altitude, altitude can also be estimated. There are several versions of this sensor module. The BME280 sensor uses the I2C or SPI communication protocol to communicate with the microcontroller. ACS712 is a fully integrated Hall-effect linear current sensor with 2.1 kV RMS voltage isolation and an integrated low-resistance current conductor [21]. It is simply presented as a current sensor that uses its conductor to calculate and measure the magnitude of the applied current. This work used an 80mm Snowman PWM adjustable fan with four pins for cooling, powered by 12 VDC.

A. Layout Description

The layout is powered by two power sources: the microcontroller is powered from 5 V via a USB cable, and the cooler is powered by an adapter with an input voltage of 12 V. The microcontroller communicates with the devices using general-purpose input pins. Interaction with the temperature sensor is performed through the I2C protocol, which implies that the microcontroller works as a control device and the temperature sensor works as an auxiliary device. The cooler is controlled by sending a PWM signal and the speed is measured using a tachometer built into the cooler. The output value of the current sensor is read through the analog input of the microcontroller. Figure 1 shows the experimental layout.

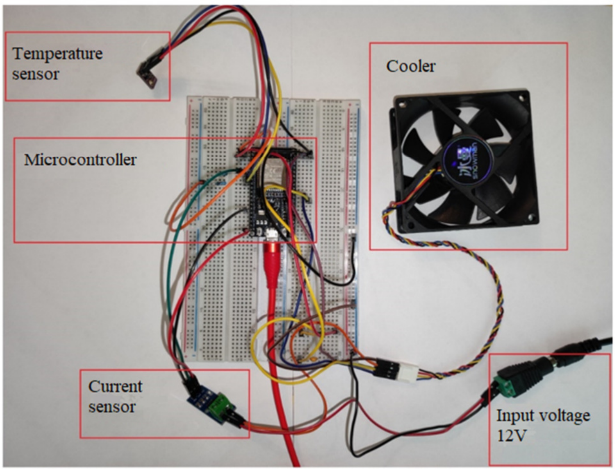

Fig. 1. Experimental layout.

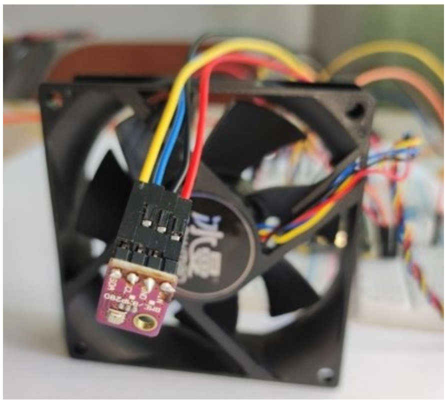

Fig. 2. Location of the temperature sensor relative to the cooler.

The temperature sensor is fixed so that the cooler can cool it as efficiently as possible, as shown in Figure 2.

B. *Algorithms Definition*

PID controllers are based on a feedback mechanism in the loop to control the process parameters. PID controllers are the most accurate and stable controllers available today. Their basic principle of operation is based on the separate adjustment or tuning of each component. A correction factor is applied to the input signal, which is calculated based on the difference between the values of the proportional, integral, and derivative components. Proportional tuning involves adjusting the target in proportion to the difference between the target and the current characteristics. In this way, the target value is never reached because, as the difference approaches zero, so does the applied correction. The error accumulated as a result of the proportional tuning is corrected by the integral tuning, which tends to increase the correction factor. For example, if the temperature remains below the set point, the integral tuning will seek to increase the supply head. However, when the setpoint temperature is reached, integral tuning will attempt to reduce the cumulative error to zero instead of stopping the heating, resulting in an overshoot. To minimize overshoot, a derivative tuning is used that seeks to reduce the correction factor applied as the target is approached.

Over the years, many rules have been proposed to address the question of how to tune a PID controller. Probably the first and certainly the best known are the Zeigler-Nichols (ZN) [22] rules, introduced in the 1940s. As in the method described above, the gain factors K_i and K_d are first set to zero. The gain is increased until it reaches a limiting value, K_u, at which the output of the circuit begins to oscillate. Table I shows K_u and the oscillation period P_u used to set the gain.

TABLE I. SETTINGS ACCORDING TO THE ZIEGLER–NICHOLS METHOD

Control type	K_p	K_i	K_d
P	0.50 K_u	-	-
PI	0.45 K_u	1.2 K_p/K_u	-
PID	0.60 K_u	2 K_p/P_u	K_p P_u/8

The PID control scheme is named after its three correction elements, the sum of which makes up the anipulated variable. The proportional, integral, and derivative components are summed to calculate the output signal of the PID controller. Defining $u(t)$ as the controller output signal, the final form of the PID algorithm is:

$$u(t) = k_p e(t) + k_i \int_0^t e(t)dt + k_d \frac{d}{dt} \quad (1)$$

where k_p is the proportional gain tuning parameter, k_i is the integral gain tuning parameter, k_d is the derivative gain tuning parameter, e is the error setting actual temperature, and t is the time or instantaneous time (present).

The coefficients K_p = 83.88, K_i = 31.24, and K_d = 50.31 of the PID controller are obtained using the Ziegler-Nichols method. The module operates as shown in Figure 3.

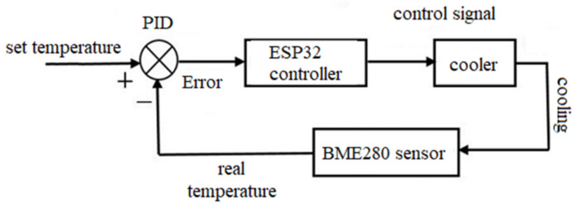

Fig. 3. PID module operation logic diagram.

Fuzzy logic controllers produce an acceptable but definite result in response to incomplete, ambiguous, distorted, or inaccurate (fuzzy) input [23]. Fuzzy logic is a reasoning method that resembles human reasoning. The fuzzy logic approach mimics the way humans make decisions, which includes intermediate possibilities between numerical values: Yes and No. A conventional logic unit, understood by a computer, takes precise input data and produces a certain output as either True or False, which is equivalent to a human Yes or No. Unlike computers, human decision-making involves a range of possibilities from Yes to No, or 1 to 0, such as definitely Yes, maybe Yes, don't know, maybe No, and definitely No. Fuzzy logic works on levels of input probabilities to achieve a certain result. As shown below, the controller consists of four main parts:

- The fuzzy logic system conversion module converts the input data of the system, which are crisp numbers, into fuzzy sets. It decomposes the input into five steps, as shown in Table II.

TABLE II. STEPS OF CONVERSION TO FUZZINESS

LP	Number × Large Positive
MP	Number × Medium Positive
S	Number × Small
MN	Number × Medium Negative
LN	Number × Large Negative

- The knowledge base stores the If-Then rules provided.
- The logical inference mechanism simulates the human thinking process by making fuzzy inferences about the input data and If-Then rules.

- The clarity conversion module converts the fuzzy set obtained by the logical inference mechanism into a crisp value.
- The membership functions quantify the linguistic term and graphically represent the fuzzy set. The membership function of a fuzzy set A in the discourse space X is defined as $\mu A:X \rightarrow [0, 1]$. Each element X is defined by a value between 0 and 1 and shows the degree of membership of the element X to the fuzzy set A. The discourse space is represented on the x-axis, and the degree of membership is represented on the y-axis. In this case, several membership functions can be used to convert to a fuzzy numerical value. The use of complex membership functions is redundant, as it does not increase the accuracy of the output. Figure 4 shows all the membership functions for LP, MP, S, MN, and LN.

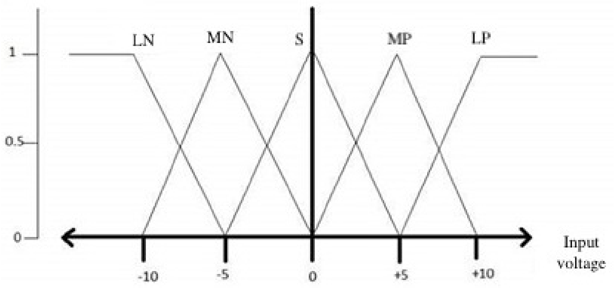

Fig. 4. Membership functions.

Triangular shapes of membership functions are most common compared to other shapes such as trapezoidal, single element, and Gaussian. Here, the input of the 5-level fuzzy converter varies from -10 to 10 V. Thus, the corresponding output also varies. The following steps describe the design of the fuzzy logic-based controller:

- Step 1 - Definition of linguistic variables and terms. Linguistic variables are input and output variables in the form of simple words or sentences. The linguistic terms for room temperature are cold, warm, heat, hot, etc. Each term in this set can cover some subset of all temperature values.
- Step 2 - Construct the membership function for the input values. Figure 5 shows the membership functions of the temperature variable.

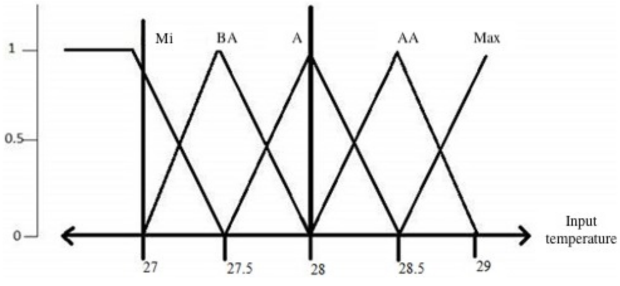

Fig. 5. Variable temperature membership functions.

Step 3 - Construct sets of output values. The output linguistic variables express the applied values to the activators of the temperature controller. This study considered one output variable: the PWM signal for the fan. In this case, it is necessary to relate the membership functions to the output variable, which should be identical to the input variable. Table III shows the set of output variables used for the PWM signal and Table IV shows the rule sets in the knowledge base in the form of If-Then-Else.

TABLE III. SET OF OUTPUT VALUES

No.	Output value range	Cooler power	Variable name
1	165.75-255	65%-100%	Maximum
2	127-204	50%-80%	Above Average
3	167.75-89.25	65%-35%	Average
4	127-51	50%-20%	Below Average
5	89.25-0	35%-0%	Minimum

TABLE IV. OUTPUT VALUE MEMBERSHIP FUNCTIONS

No.	Rule name	Variable in effect
1	If temperature=Max Then	Maximum
2	If temperature=AA Then	Above Average
3	If temperature=A Then	Average
4	If temperature=BA Then	Below Average
5	If temperature=Min Then	Minimum

- Step 5 - Obtain an imprecise value. Inexact set operations perform rule evaluation. The operations used for OR and AND are Max and Min, respectively. All evaluation results are combined to form the required imprecise value.
- Step 6 - Output the exact value. This is performed according to the membership function for the output variable, as shown in Figure 6.

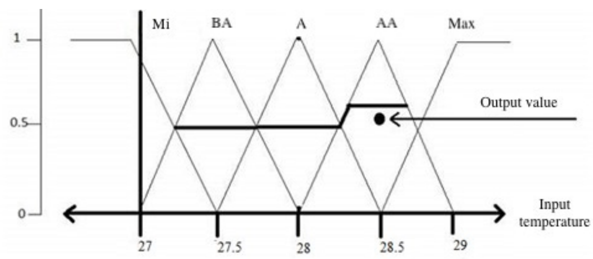

Fig. 6. The exact value display.

III. RESULTS AND DISCUSSION

An experimental study was conducted using the previously described system, including three stages: PID, FLC, and combined method. Its purpose was to identify the most energy-efficient algorithm. The components were: heater, cooler, temperature sensor, and power consumption sensor. The heater was a Polaris electric radiator with heating control.

The following steps were the same for all stages: start the cooler, wait for the system to stabilize, start the timer, start the heater, stop the timer, and receive data. These steps ensure the same conditions, minimizing the influence of external factors. Once the system is stabilized, the timer and the heater are started. After a certain time, the timer stops without taking further data from the temperature sensor. The resulting data are displayed in the form of graphs. Each step will go through several iterations to get the average values. The results are presented in the form of tables and comparison graphs. The tables contain the average stabilization time and average power consumption, calculated over 10 iterations. When the number of iterations was increased beyond 10 (the maximum value was 70), no significant changes were observed in the mean values of rise time, stabilization time, and power consumption, therefore, it was decided to use 10 iterations. The graphs show the state of the system during the allotted time interval of 45 s.

A. PID

This stage examined the previously discussed PID algorithm. Algorithm 1 describes the PID algorithm.

```
Algorithm 1: PID-controller algorithm
Kp ← some_kp
Ki ← some_ki
Kd ← some_kd
setpoint ← setpoint
input ← _input
output ← _output
delta_input ← prev_input - input
prev_input ← input
error ← setpoint - input
output ← error * Kp
output ← output + delta_input * Kd / _dt_s
if pid_integral_window > 0 then
  if f(++t >= pid_integral_window) then
    t ← 0
  integral ← integral - errors[t]
  errors[t] ← error * Ki * _dt_s
  integral ← integral + errors[t]
  end if
else
  integral ← integral + error * Ki * _dt_s
end if
integral ← integral + delta_input * Kp
integral ← constrain(integral, min_val, max_val)
output ← output + integral
output ← constrain(output, min_val, max_val)
```

Part of the study was performed manually, that is, bringing the electric heat sink closer and farther away from the temperature sensor to induce a temperature spike in the system. The study went through 10 iterations and average values were taken. Table V shows the stabilization time and the average power consumption of the system. Figure 7 shows the behavior of the system during the allotted time interval of 45 s.

TABLE V. PID RESULTS

Iteration number	Rise time (s)	Stabilization time (s)	Power consumption (W)
1	9.68	27.43	0.86
2	9.55	27.31	0.87
3	10.4	27.92	0.9
4	8.92	26.1	0.72
5	9.4	27.12	0.82
6	10.3	27.8	0.88
7	9.32	26.71	0.79
8	9.1	26.5	0.75
9	9	26.4	0.74
10	9.85	27.7	0.87
Average	**9.55**	**27.1**	**0.82**

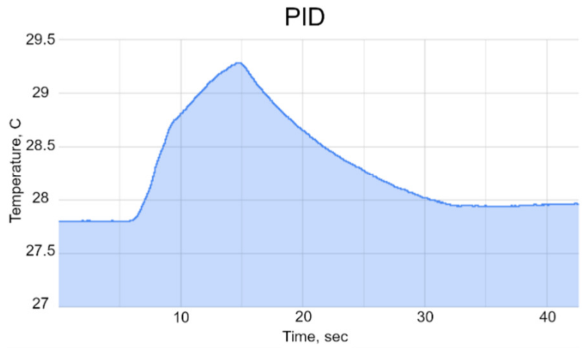

Fig. 7. PID research.

The PID-regulator reacts to a sudden change in temperature nonlinearly, due to the constant calculation of the error. It should also be noted that the temperature output over the interval is larger due to the gradual accumulation of the proportional component of the algorithm. As soon as the peak temperature value is overcome, the algorithm smoothly returns the temperature to the set value, which in this case is 28°C.

B. FLC Study

This stage examined the FLC algorithm, as shown in Algorithm 2. This study took 10 iterations, and the results are shown in Table VI and Figure 8.

```
Algorithm 2 FLC algorithm
temperature ⟵ cold, safe, hot
speed ⟵ slow, average, fast
fuzzyRules.add(1, ifTemperatureCold, thenSpeedSlow)
fuzzyRules.add(2, ifTemperatureSafe, henSpeedAverage)
fuzzyRules.add(3, ifTemperatureHot, thenSpeedFast)
input ⟵ getActualTemperature()
fuzzify(input)
output ⟵ deffuzify()
updateFanSpeed(output)
```

TABLE VI. FLC RESULTS

Iteration number	Rise time (s)	Stabilization time (s)	Power consumption (W)
1	10.95	39.1	0.69
2	9.62	38.2	0.68
3	10.51	37.06	0.63
4	9.89	38.82	0.71
5	11.45	37.85	0.63
6	9.65	38.25	0.64
7	10.55	39.5	0.73
8	9.52	37.6	0.62
9	11.37	38.92	0.72
10	11.16	39.7	0.75
Average	**10.5**	**38.5**	**0.68**

As can be seen in Figure 8, the allotted time to stabilize the system is not enough for FLC, which is its disadvantage. However, it has low power consumption because of the small number of calculations and the small fluctuations in the control signal. The results can be improved by resizing the fuzzy sets and fine-tuning the membership functions. Nevertheless, this requires increasing the requirements on computing resources, which contradicts the objective.

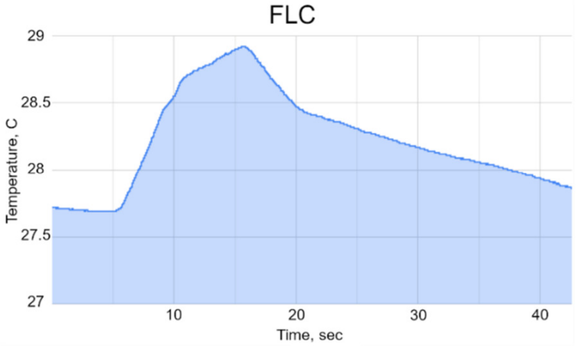

Fig. 8. FLC study.

C. Study of the Combined Method

This stage examined the combined method of the two algorithms. Algorithm 3 shows the pseudocode of the combined method.

```
Algorithm 3: PID_FLC algorithm
input ← getActualTemperature()
if (input >2 7.0 & input <= 28.0 ) then
  fuzzyControl()
else if (input > 28) then
  while (input >= 27.5) do
    pidControl()
    updateBME280()
    getAmpsWatts()
  end while
end if
```

Since temperature holding assumes an interval of permissible values, the advantage of FLC here is unambiguous. Hence, it can be used when the temperature is within the permissible interval. As soon as the temperature is outside the interval, PID has the fastest response to the event but works with an error: the set temperature, of which there are several in the interval. Thus, for the set temperature, the median of the interval of valid values is taken. FLC works when the temperature is within the acceptable interval, and PID works when it goes outside the limits. The study was carried out in 10 iterations, and Table VII and Figure 9 show the results.

TABLE VII. COMBINED METHOD RESULTS

Iteration number	Rise time (s)	Stabilization time (s)	Power consumption (W)
1	9.1	22.41	0.54
2	8.43	21.8	0.51
3	9.6	22.73	0.56
4	9.05	22.21	0.54
5	10.3	23.9	0.58
6	8.2	21.7	0.5
7	8.71	22.3	0.54
8	10.2	23.86	0.58
9	9.13	22.23	0.54
10	8.8	21.9	0.51
Average	9.15	22.5	0.54

Figure 9 shows that the rise time is about half the system stabilization time, which indicates its efficiency. A slight decrease towards the end of the graph demonstrates the switch of system control from PID to FLC. The explanation for the rapid response of PID to temperature outside the interval is the transmission of an already high control signal by the FLC algorithm, which eliminates the accumulation time of the proportional component of PID. Figure 10 shows a comparison of the combined method with the PID and FLC algorithms. The experimental results show that the combined method reduced the rise time by at least 5%, stabilization time by at least 17%, and power consumption by at least 21%.

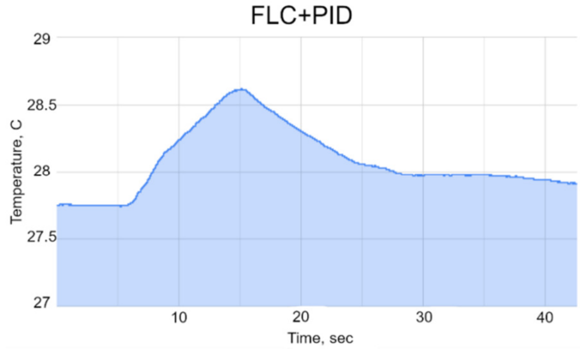

Fig. 9. Combined method.

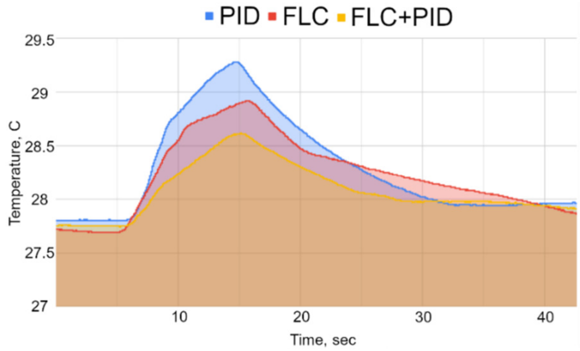

Fig. 10. Method comparison.

IV. CONCLUSION

This study describes the design and implementation of a cooling system to examine the energy efficiency capabilities of different algorithms. Two algorithms were selected, PID and FLC. A combined method was also proposed, which involves utilizing only the strengths of each algorithm. A model consisting of a temperature sensor, a cooler, a current sensor, and a microcontroller was assembled. The study took place in three stages, testing each of the algorithms individually. The evaluation criteria were system stabilization time and cooler power consumption during the time interval. According to the results, the proposed combined method was the more efficient.

This study differs from previous ones in the transparency of the temperature parameter setting algorithm and by combining the conventional PID and the fuzzy PID controllers. Combining the two methods reduced power consumption by 21%, rise time by 5%, and stabilization time by 17%. It should be noted that the combined method can be further improved by changing the size of fuzzy sets and by more accurately tuning the membership functions in the FLC algorithm. This algorithm and the assembled layout have application limitations, as the sampling period of the sensors must be higher than the frequency of the control signals. In summary, the proposed method minimizes the total cost of hardware and software development, whereas its application can bring both economic and environmental benefits to data centers.

REFERENCES

[1] P. Cominos and N. Munro, "PID controllers: recent tuning methods and design to specification," *IEE Proceedings - Control Theory and Applications*, vol. 149, no. 1, pp. 46–53, Jan. 2002, https://doi.org/10.1049/ip-cta:20020103.

[2] S. K. Pandey, K. Veeranna, B. Kumar, and K. U. Deshmukh, "A Robust Auto-tuning Scheme for PID Controllers," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, Singapore, Oct. 2020, pp. 47–52, https://doi.org/10.1109/IECON43393.2020.9254382.

[3] W. W. Shein, Y. Tan, and A. O. Lim, "PID Controller for Temperature Control with Multiple Actuators in Cyber-Physical Home System," in *2012 15th International Conference on Network-Based Information Systems*, Melbourne, Australia, Sep. 2012, pp. 423–428, https://doi.org/10.1109/NBiS.2012.118.

[4] N. H. A. Hamid, M. M. Kamal, and F. H. Yahaya, "Application of PID controller in controlling refrigerator temperature," in *2009 5th International Colloquium on Signal Processing & Its Applications*, Kuala Lumpur, Malaysia, Mar. 2009, pp. 378–384, https://doi.org/10.1109/CSPA.2009.5069255.

[5] Y. Hu, "MCU-Based PID Temperature Control System for Linear Heating and Cooling," *Academic Journal of Science and Technology*, vol. 8, no. 1, pp. 212–215, Nov. 2023, https://doi.org/10.54097/ajst.v8i1.14313.

[6] M. Khalil, A. S. McGough, Z. Pourmirza, M. Pazhoohesh, and S. Walker, "Machine Learning, Deep Learning and Statistical Analysis for forecasting building energy consumption — A systematic review," *Engineering Applications of Artificial Intelligence*, vol. 115, Oct. 2022, Art. no. 105287, https://doi.org/10.1016/j.engappai.2022.105287.

[7] M. Anastasiadou, V. Santos, and M. S. Dias, "Machine Learning Techniques Focusing on the Energy Performance of Buildings: A Dimensions and Methods Analysis," *Buildings*, vol. 12, no. 1, Jan. 2022, Art. no. 28, https://doi.org/10.3390/buildings12010028.

[8] R. Patel and V. Kumar, "Multilayer Neuro PID Controller based on Back Propagation Algorithm," *Procedia Computer Science*, vol. 54, pp. 207–214, Jan. 2015, https://doi.org/10.1016/j.procs.2015.06.023.

[9] M. Ray, P. Samal, and C. K. Panigrahi, "Implementation of a Hybrid Technique for the Predictive Control of the Residential Heating Ventilation and Air Conditioning Systems," *Engineering, Technology & Applied Science Research*, vol. 12, no. 3, pp. 8772–8776, Jun. 2022, https://doi.org/10.48084/etasr.5027.

[10] A. Marvuglia, A. Messineo, and G. Nicolosi, "Coupling a neural network temperature predictor and a fuzzy logic controller to perform thermal comfort regulation in an office building," *Building and Environment*, vol. 72, pp. 287–299, Feb. 2014, https://doi.org/10.1016/j.buildenv.2013.10.020.

[11] J. C. Mugisha, B. Munyazikwiye, and H. R. Karími, "Design of temperature control system using conventional PID and Intelligent Fuzzy Logic controller," in *2015 International Conference on Fuzzy Theory and Its Applications (iFUZZY)*, Yilan, Taiwan, Nov. 2015, pp. 50–55, https://doi.org/10.1109/iFUZZY.2015.7391893.

[12] H. Yan, Y. Xia, X. Xu, and S. Deng, "Inherent operational characteristics aided fuzzy logic controller for a variable speed direct expansion air conditioning system for simultaneous indoor air temperature and humidity control," *Energy and Buildings*, vol. 158, pp. 558–568, Jan. 2018, https://doi.org/10.1016/j.enbuild.2017.10.013.

[13] A. Chojecki, A. Ambroziak, and P. Borkowski, "Fuzzy Controllers Instead of Classical PIDs in HVAC Equipment: Dusting Off a Well-Known Technology and Today's Implementation for Better Energy Efficiency and User Comfort," *Energies*, vol. 16, no. 7, Jan. 2023, Art. no. 2967, https://doi.org/10.3390/en16072967.

[14] J. Gonzalez-Villagomez, C. Rodriguez-Donate, M. Lopez-Ramirez, R. I. Mata-Chavez, and O. Palillero-Sandoval, "Novel Iterative Feedback Tuning Method Based on Overshoot and Settling Time with Fuzzy Logic," *Processes*, vol. 11, no. 3, Mar. 2023, Art. no. 694, https://doi.org/10.3390/pr11030694.

[15] M. M. Rahman and M. S. Islam, "Design of a Fuzzy Based Pid Algorithm for Temperature Control of An Incubator," *Journal of Physics: Conference Series*, vol. 1969, no. 1, Apr. 2021, Art. no. 012055, https://doi.org/10.1088/1742-6596/1969/1/012055.

[16] L. I. Minchala, J. Peralta, P. Mata-Quevedo, and J. Rojas, "An Approach to Industrial Automation Based on Low-Cost Embedded Platforms and Open Software," *Applied Sciences*, vol. 10, no. 14, Jan. 2020, Art. no. 4696, https://doi.org/10.3390/app10144696.

[17] Z. Yu, N. Liu, K. Wang, X. Sun, and X. Sheng, "Design of Fuzzy PID Controller Based on Sparse Fuzzy Rule Base for CNC Machine Tools," *Machines*, vol. 11, no. 1, Jan. 2023, Art. no. 81, https://doi.org/10.3390/machines11010081.

[18] K. A. Al Sumarmad, N. Sulaiman, N. I. A. Wahab, and H. Hizam, "Energy Management and Voltage Control in Microgrids Using Artificial Neural Networks, PID, and Fuzzy Logic Controllers," *Energies*, vol. 15, no. 1, Jan. 2022, Art. no. 303, https://doi.org/10.3390/en15010303.

[19] "ESP-IDF Programming Guide - ESP32." [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/.

[20] "Adafruit BME280 Library 1.0 documentation." https://docs.circuitpython.org/projects/bme280/en/latest/.

[21] Allegro Microsystems Inc., "ACS712 - Fully Integrated, Hall Effect-Based Linear Current Sensor with 2.1 kVRMS Voltage Isolation and a Low-Resistance Current Conductor." [Online]. Available: https://www.sparkfun.com/datasheets/BreakoutBoards/0712.pdf.

[22] J. G. Ziegler and N. B. Nichols, "Optimum Settings for Automatic Controllers," *Journal of Dynamic Systems, Measurement, and Control*, vol. 115, no. 2B, pp. 220–222, Jun. 1993, https://doi.org/10.1115/1.2899060.

[23] V. N. Alexandrov, J. J. Dongarra, B. A. Juliano, R. S. Renner, and C. J. K. Tan, Eds., "Computational Science - ICCS 2001 Proceedings, Part II," San Francisco, CA, USA, May 2001, https://doi.org/10.1007/3-540-45718-6.