INTERNATIONAL JOURNAL OF INTEGRATED ENGINEERING
ISSN: 2229-838X e-ISSN: 2600-7916

Vol. 16 No. 5 (2024) 359-372
https://publisher.uthm.edu.my/ojs/index.php/ijie

IJIE

Design and Implementation of Fuzzy-based Fine-tuning PID Controller for Programmable Logic Controller

Van-Khanh Nguyen[1]*, Vy-Khang Tran[1], Hai Pham[1,2], Hoang-Dung Nguyen[1], Chi-Ngon Nguyen[1]

[1] *Faculty of Automation Engineering, College of Engineering,*
Can Tho University, Campus II, 3/2 street, Ninh Kieu district, Can Tho city, 94000, VIETNAM
[2] *Aerospace Engineering and Aviation Discipline,*
School of Engineering, RMIT University, GPO Box 2476, Melbourne, VIC 3001, AUSTRALIA

*Corresponding Author: vankhanh@ctu.edu.vn

DOI: https://doi.org/10.30880/ijie.2024.16.05.027

Article Info

Received: 17 January 2024
Accepted: 21 June 2024
Available online: 12 August 2024

Keywords

Fuzzy control, programmable control, automatic control, industrial control

Abstract

The Proportional-Integral-Derivative (PID) controller, already known for its stability, is widely used in industrial applications and integrated into many Programmable Logic Controllers (PLCs). However, most PLCs do not support the self-tuning mechanism for PID controller parameters. Therefore, users must manually adjust several times to achieve the desired outcomes. This manual adjustment is time-consuming and must be repeated as control object parameters change over time. This study proposed a fine-tuning mechanism for the PID controller's parameters based on a fuzzy-PD controller. The mechanism was designed and simulated using MATLAB/Simulink on an identified plant, then converted into a Structured Control Language (SCL) code for implementation on the PLC programs. Experimental results on the Siemens S7-1200 PLC demonstrated the proposed mechanism's effectiveness in stabilizing the thermal plant by adjusting the initial parameters of the integrated PID controller. The system response was more stable, and the overshoot was minimized in comparison with the built-in auto-tuning feature on the S7-1200. Specifically, overshoot decreased to 0.79% from 0.94%, and the setting error declined to 0.1 °C from 0.45 °C. The above results indicate the effectiveness of the proposed self-tuning mechanism when used to improve the quality of PID controllers in PLCs. In addition, due to its ability to self-tuning parameters, it helps users reduce the time required to design PID controllers.

1. Introduction

The Proportional-Integral Derivative (PID) controller is a classic control algorithm that has been widely applied in many industrial control systems in different fields such as energy, temperature control, and manufacturing [1–5]. The PID controller integrated into Programmable Logic Controllers (PLCs) in control and automation applications is becoming increasingly popular. This controller on PLCs has been applied in many fields such as industry [6, 7], motor speed control [8- 10], and production systems [11]. However, PID controllers are often designed and tuned for best results before being implemented into a PLC. This design process takes a long time to achieve the expected results. Therefore, an auto-tune mechanism needs to be designed and implemented on the PLCs.

MATLAB/Simulink is a well-known tool for designing and simulating control systems [12]. It is also considered a support tool for implementing new controllers in PLCs. Saputra et al. used Matlab/Simulink to identify, design, and simulate a DC motor control system and apply a PID controller to the PLC [8]. Kula et al. [13] designed and simulated a PID controller to control rectification column based on a PLC using MATLAB/Simulink. Osman et al. [14] developed a building control system using the MATLAB/Simulink tool; the author used Simulink to build a mathematical model that can assist in the design of the control system. Wei used MATLAB/Simulink to design an adaptive fuzzy-PID controller to monitor the solar power system [15]. Dettori et al. also used MATLAB/Simulink to design a fuzzy logic PID controller to control steam turbines in solar applications [16]. Mathwork has developed the PLC Coder Toolbox to support the conversion of Simulink blocks to Structured Control Language (SCL) code to execute them on the PLC [17]. Algburi uses Matlab to design the fuzzy controller in the temperature control system and uses PLC Coder to convert it to SCL to deploy it on the PLC [18]. PLC Coder Toolbox can convert blocks on Simulink to SCL functions so that it can be easily integrated into PLCs. This tool is useful for implementing new features or controllers in PLCs. MATLAB also integrates a system identification tool that effectively supports the identification of mathematical equations for control plants, which is very important information for the design and simulation of control structures. The application of this tool in the design of new controllers for PLCs should be considered.

With the development of fuzzy logic and artificial neural network theory, the integration of intelligent control into control systems has become an important research field [19]. Fuzzy controllers are used in many control systems to improve the performance and accuracy of control processes. In addition, the fuzzy controller has been used to automatically adjust the parameters K_P, K_I, and K_D of the PID controller to adapt to the control plant [7, 8, 15, 20, 21, 22]. The combination of fuzzy theory and PID contributes to increasing the efficiency and accuracy of the control process, particularly in cases where the control plant has characteristics that change over time. However, these methods could cause instability in the system because the fuzzy controller directly adjusted the parameters of the PID controller instead of simply tuning them in constrained ranges. Therefore, a method to fine-tune the parameters of the PID controller based on fuzzy logic is proposed to overcome these limitations.

This paper proposes a simple and fast process for designing and implementing self-tuning PID controllers on PLCs based on fuzzy-PD controllers. The fuzzy-PD controller is a fuzzy controller designed based on the error and derivative error of the control system to determine the rules for fine-tuning the coefficients of the target PID controller. MATLAB/Simulink was used to identify the transfer function (TF) of the control plant based on its acquired data. A PID controller with non-optimal initial coefficients was applied to control the identified plant. The mechanism for fine-tuning these parameters is designed and simulated based on the fuzzy-PD controller with the recognition plant in MATLAB/Simulink. The accepted mechanism is then converted to the SCL to be directly implemented on the Siemens PLC. The outcome of this research helps to reduce the time needed to design and adjust the PID controller on the PLCs to adapt to the changing of the plant's parameters during its operation. Moreover, this method is not only applicable to Siemens PLCs but also to others.

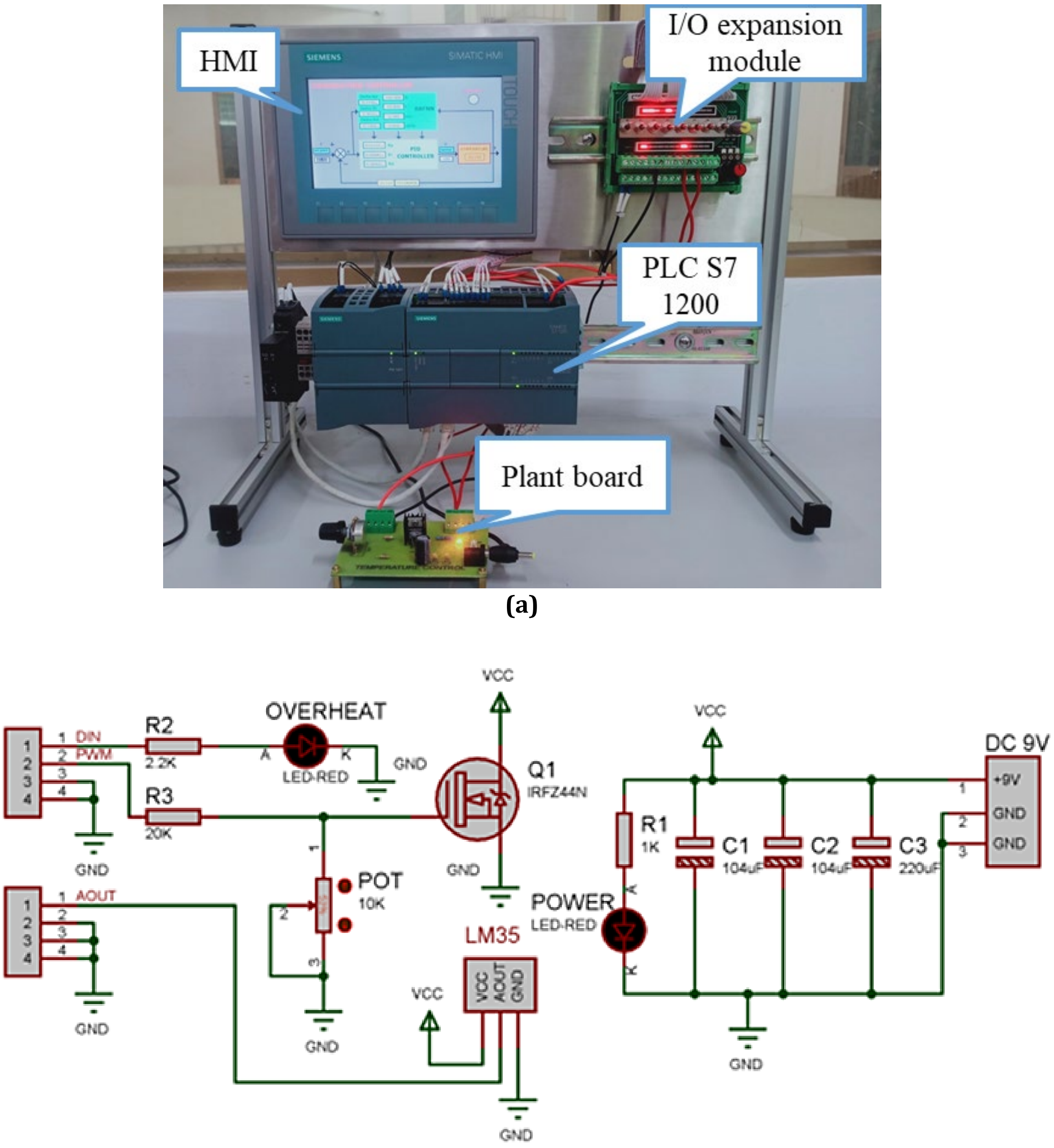
2. Methodology

2.1 Overview

This study designed and integrated a mechanism to automatically fine-tune the parameters of the PID controller into the PLC. First, the MATLAB identification toolbox (MSIT) was used to identify the mathematical model of the control object. Next, the fine-tuning mechanism was designed based on the fuzzy-PD controller and simulated using MATLAB/Simulink with the PID controller and a recognition plant. The fine-tuning mechanism was then converted to SCL code using the PLC coder [20], so that it could be deployed on the S7-1200 PLC. On the PLC, the parameters of the PID Compact, an integrated PID controller function block (FB), were adjusted using the proposed mechanism to control the temperature of the real plant. The built-in auto-tuning feature of TIA Portal software (programming integration platform for the Siemens PLC) is also applied to compare and evaluate the proposed mechanism.

2.2 Experimental Setup

This study used a control plant designed based on the TCLab kit [23], as presented in Fig. 1a. The control plant on the TCLab kit uses the working temperature of a bipolar junction transistor (BJT), which the bias current at the B gate controls [24]. In this study, an N-channel MOSFET (Q1) was proposed to replace the BJT to simplify the bias design. A schematic of the new thermal plant is shown in Fig. 1. The pulse width modulation (PWM) technique was used to convert the control law into a bias voltage at the G gate of Q1 to control its temperature. The maximum bias voltage of the G gate is 2 V to ensure that Q1 did not overheat because no load was applied at the D gate. The temperature of Q1 was fed back to the controller by an LM35 thermal sensor in direct contact with Q1. A variable resistor is also used to adjust the bias voltage at the G gate of Q1.



(a)

(b)

Fig. 1 *(a) The experimental model setup; and (b) The control plant schematic*

2.3 Plant Identification

MSIT is a powerful system identification tool. This toolbox is particularly useful for estimating parameters or identifying mathematical models of control systems, integrating many methods for identifying nonlinear systems [25]. Estimation data are usually collected from experiments on the control plant, typically the input and output of the plant. The MSIT uses these data to identify a plant's model or parameters. The identified model is then used to design and simulate the controllers [26]. In this study, the identification model is used to simulate and test the fine-tuning mechanism of the PID controller's coefficients before implementing it on the target PLC.

2.4 Fuzzy-PD Controller Based Fine-Tuning Mechanism

Current fuzzy-PID controllers directly adjust the PID parameters based on the error between the set value and feedback value, along with the rate of change based on fuzzy inference [19]. This study also proposes a mechanism to adjust the PID coefficients using the fuzzy-PD controller, but it does not directly adjust the coefficients but only fine-tunes them. This method ensures system stability because the main control energy still comes from the PID controller. This is particularly relevant when applied to industrial control systems using PLCs, where control stability is a priority. The design process starts by adjusting the parameters of the PID so that the plant can initially track the set value. Subsequently, the proposed fine-tuning mechanism optimizes these parameters. The proposed fine-tuning structure and its application to the integrated PID of PLC S7-1200 are illustrated in Fig. 2. With the designed thermal plant, only two parameters, K_P and K_D, were adjusted, and K_I was kept fixed. The proposed tuning mechanism is implemented and simulated in advance with the identified plant and the PID controller in MATLAB/Simulink. It is then converted to the SCL for execution on the S7-1200. The moving average filter is used to smooth the feedback signal to limit the chattering phenomenon of the control signal that may affect the control plant in some cases. The mathematical equation of the moving average filter is shown in Eq. (1):

$$y(n)=\frac{1}{M}\sum_{i=0}^{M-1}x(n+i) \tag{1}$$

Where *y(n)* is the filtered feedback signal, *x* is the feedback signal, and *M* is the window size.

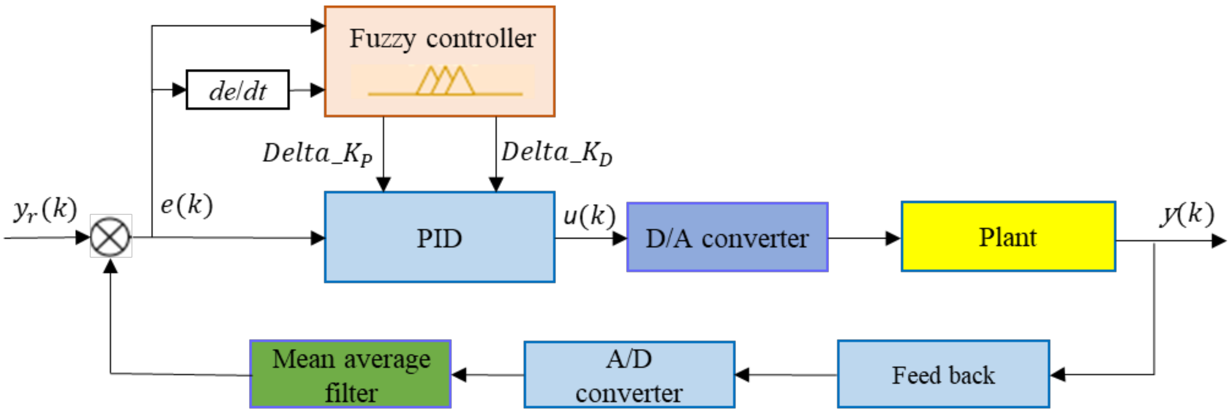


Fig. 2 *PID controller with the proposed fine-tuning mechanism*

On S7-1200, the PID controller is a built FB, and the algorithm is performed according to the following Eq. (2) [27]:

$$Y=k_p[\left(b*w*(-x)\right)+\frac{1}{T_I*s}(w-x)+\frac{T_D*s}{a*T_D*s+1}(b*w-x)] \tag{2}$$

Where, *Y* is the controller output value, K_p is the proportional gain, T_I is the integral phase sampling time, and T_D is derivative time; *a* is the Hysteresis coefficient of the *D*, *b* is the weight of the stitch *P*, *c* is the weight of the stitch *D*, and *s* is the Laplace operator; *w* is the set-point value and *x* is the process value.

The TIA Portal integrates an auto-tuning tool to quickly determine the coefficients for the PID Compact. In this study, to evaluate the effectiveness of the proposed fine-tuning mechanism, the PID parameters were first manually adjusted based on the Ziegler-Nichols method [28]. The coefficients are tuned such that the temperature of the plant initially tracks the set value, and the response may remain fluctuating. The coefficients were then fine-tuned using the proposed method to improve the control quality. As mentioned, the fine-tuning algorithm based on the fuzzy-PD controller is designed and tested using MATLAB/Simulink, with the thermal plant identified from the acquired data of the real plant and the PID controller block of MATLAB/Simulink. To synchronize during deployment on the PLC, the PID controller in MATLAB/Simulink is also manually adjusted using the Ziegler-Nichols method and then optimized using the proposed mechanism. In this study, the S-curve response analysis method was applied to determine the initial values of PID controller coefficients [29].

In this work, the fuzzy-PD controller plays a role in finding the adjustment rule for the PID controller parameters based on the system response. It is a Multiple-Input Multiple-Output (MIMO) system with two inputs that are the error *e* and derivative error *de*, and two outputs that are the adjustment amounts for the two parameters K_P and K_D. The K_I parameter is constant. The fuzzy controller uses the Mamdani inference system [28], as shown in Fig. 3.

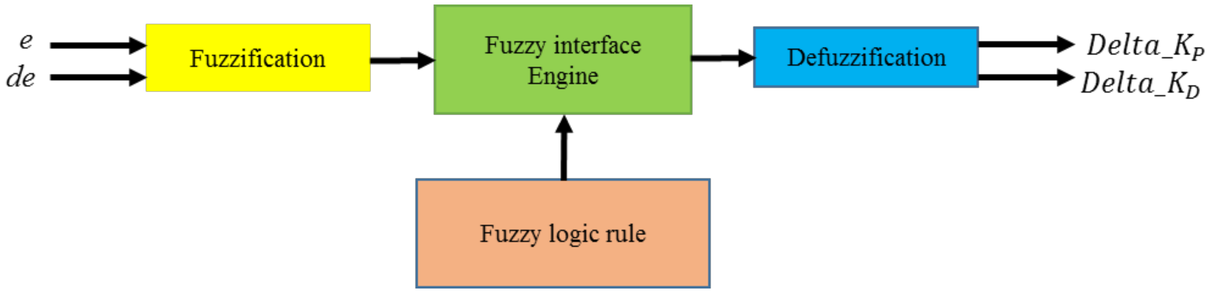


Fig. 3 *Fuzzy-PD-based self-tuning architecture*

The fuzzy controller is designed through three main stages. The early stage is fuzzification, which converts the real values of *e* and *de* to corresponding fuzzy sets through membership functions. In this study, all inputs and outputs of the fuzzy controller used three membership functions to guarantee that the converted controller fits well on the S7-1200. The membership functions of the input and output fuzzy sets are shown in Fig. 4. The fuzzy variable *e* is divided into 3 fuzzy sets, including N (negative), Z (zero), and P (positive), with fuzzy domains in the range of [−100, 100]. The fuzzy variable *de* is also segmented into three fuzzy sets, namely D (decrease), M (maintain), and I (increase). These sets have fuzzy domains that range from -1 to 1, or they correspond to the rate of change of *e*. During the sampling period, this rate of change will be ±1 °C. For the outputs, three fuzzy sets, including S (small), M (medium), and B (big), were applied for all $Delta\_K_P$ and $Delta\_K_D$, but there was a difference in fuzzy range [-0.01, 0.01] and [-2, 2], respectively. These values were selected based on the specifications of the plant and the *try-and-error* method. All membership functions are triangles represented by Eq. (3).

$$\mu(x)=\begin{cases} 0 & if \quad x\le a \\ \dfrac{x-a}{b-a} & if \quad a<x\le b \\ \dfrac{c-x}{c-b} & if \quad b<x\le c \\ 0 & if \quad x\ge c \end{cases} \tag{3}$$

Where *a*, *b*, and *c* are three points in the triangler function, as marked in Fig. 4b. The system sampling time is chosen to be 0.1 second because the plant's temperature is changing fast. The second stage is inference, in which fuzzy rules for output variables are built. Tables 1 and 2 indicate the rules of the $Delta\_K_P$ and $Delta\_K_D$, respectively. The Mamdani-type fuzzy rules have the structure shown in Eq. (4) and are the most used for the simulation and implementation of a fuzzy controller.

$$IF \quad x \quad is \quad A \quad AND \quad y \quad is \quad B \quad THEN \quad z \quad is \quad C \tag{4}$$

Where *x* and *y* are the input variables (*e* and *de*), *z* is the output variable ($Delta\_K_P$ or $Delta\_K_D$), *A* and *B* are fuzzy sets of the input variables, and *C* is a fuzzy set of the output variable. The fuzzy rules can be determined by using the person's knowledge or by simulating the process. In this work, both methods were flexibly applied. The membership function of the input variable is determined. This is accomplished by employing the min-max inference method, a characteristic of the Mamdani type. The specifics of this method are delineated in Eq. (5).

$$\begin{cases} \mu_{Delta\_K_P}(x)=\min(\mu_e(x),\mu_{de}(x)) \\ \mu_{Delta\_K_D}(x)=\min(\mu_e(x),\mu_{de}(x)) \end{cases} \tag{5}$$

Where $\mu_e(x)$ and $\mu_{de}(x)$ are the membership functions of the input *e* and *de*, respectively; and $\mu_{Delta\_K_P}(x)$ and $\mu_{Delta\_K_D}(x)$ is the membership function of the output $Delta\_K_P$ and $Delta\_K_D$.

The final stage is defuzzification, which combines all membership functions to determine a real value, which represents the output fuzzy sets. In this research, the defuzzification values for $delta\_K_P$ and $delta\_K_D$ are calculated using Eq. (6). This equation employs the Centroid of Gravity (COG) method [28].

$$defuzzification = \frac{\sum_{i=1}^{m} \mu(x_i) x_i}{\sum_{i=1}^{m} u(x_i)} \tag{6}$$

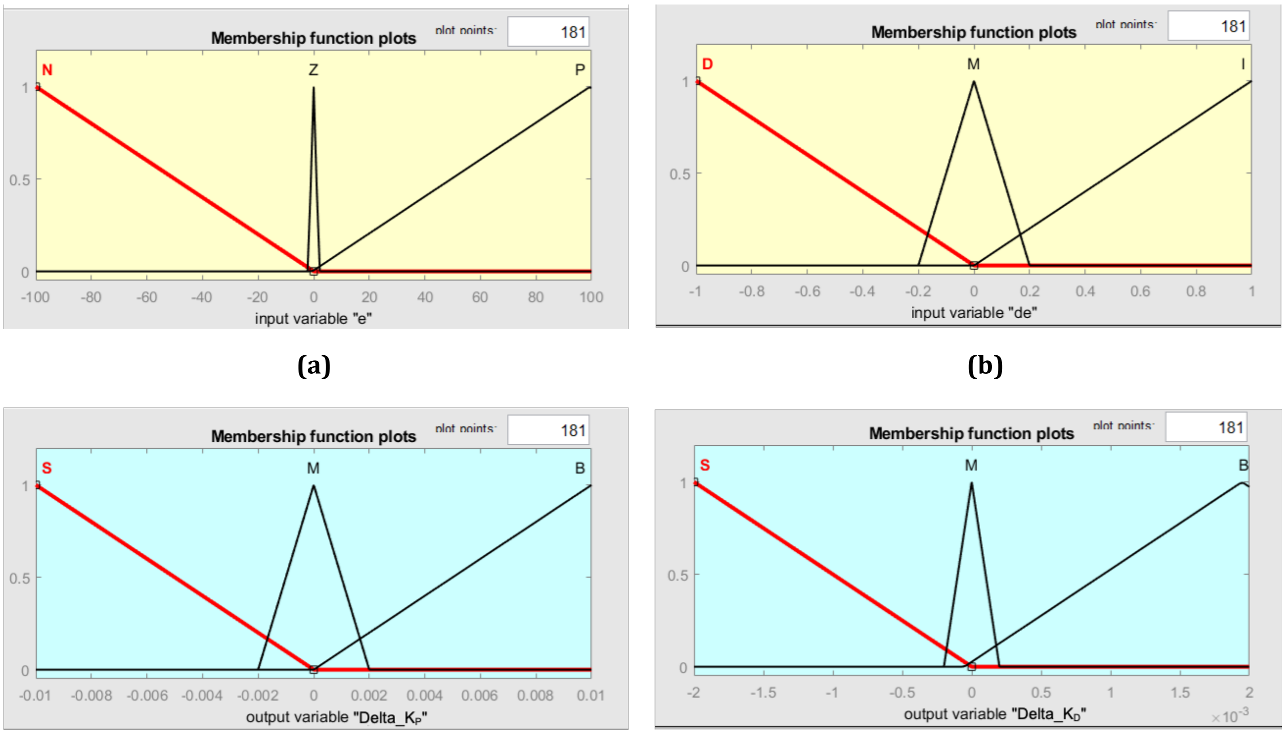
Where x_i are values of the output variable (*Delta_K*$_P$ or *Delta_K*$_D$), which are found within the fuzzy sets obtained for defuzzification, and $\mu(x_i)$ are the membership functions of x_i. All stages of the fuzzy controller were implemented in MATLAB.

Table 1 *Refined fuzzy rules of Delta_K*$_P$

		e		
		N	**Z**	**P**
	D	S	B	B
de	**M**	S	M	B
	I	S	M	B

Table 2 *Refined fuzzy rules of Delta_K*$_D$

		e		
		N	**Z**	**P**
	D	S	B	B
de	**M**	M	M	M
	I	M	S	S



(a) **(b)**

(c) **(d)**

Fig. 4 *Membership functions of a) error; b) derivative error; c) Delta_K*$_P$*; and d) Delta_K*$_D$

3. Results and Discussion

3.1 Plant Identification

The data acquired from the input and output of the control plant are depicted in Fig. 5. The data, collected over a span of 8,900 seconds, was utilized to estimate the transfer function. These data were derived from experiments conducted on the control plant, wherein the input was a step function exhibiting changes.

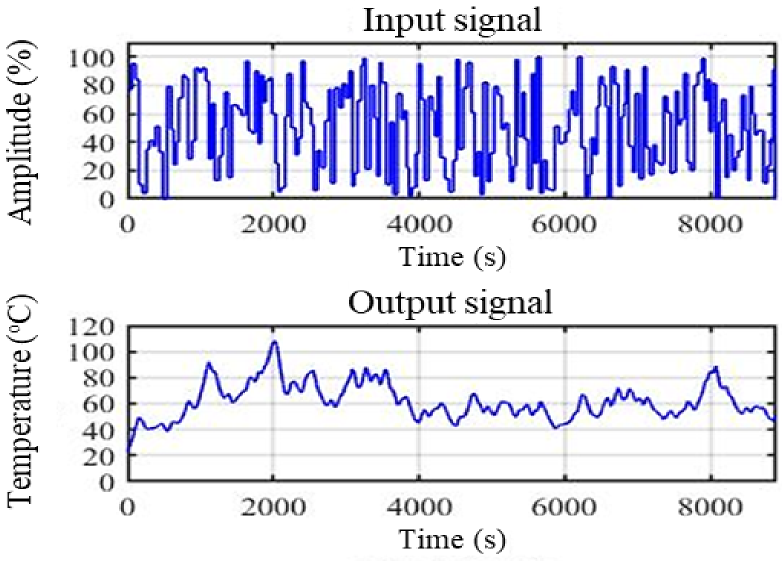


Fig. 5 *Acquired input and output data of the temperature plant*

Eq. (7) is the identified transfer function (TF) of the control plant, which was derived from the "MATLAB identification toolbox". The chart in Fig. 6 shows the temperature measured from the plant and derived by the identified TF with the same input. The results indicated that the accuracy of this TF is 81.49%. This TF was used to simulate the proposed fine-tuning algorithm before deployment on the PLC.

$$G(s)=\frac{0.003925s^{3}+1.733e-0.5s^{2}+3.79e-0.8s+2.697e-11}{s^{4}+0.005727s^{3}+2.54e-0.5s^{2}+2.973e-0.8s+2.201e-11}e^{-4s} \tag{7}$$

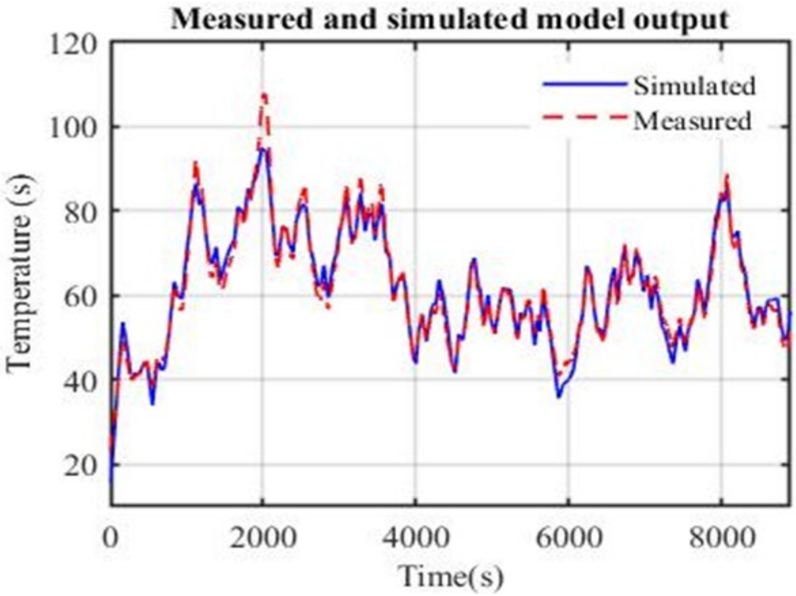


Fig. 6 *Measured and predicted data of the temperature plant*

3.2 Self-tuning PID Controller

3.2.1 Simulation of the Fine-tuning Mechanism Based on the Recognition Plant

Fig. 7 shows a Simulink diagram of the fine-tuning mechanism based on the identified plant. This diagram includes a *PID controller* block, a *Fine-tuning mechanism* block that fine-tunes the PID coefficients, and an *Identified plant*. In this study, K_I is fixed, and K_P and K_D are fine-tuned. The initial parameters of the PID controller are determined based on the Ziegler-Nichols method. Blocks SET 1, 2, and 3 were used to create the reference values for the controller. Blocks SET 4 and 5 are responsible for creating the initial K_P and K_D values. A *Noise* block is added to the feedback signal to simulate the measurement error of the feedback signal.

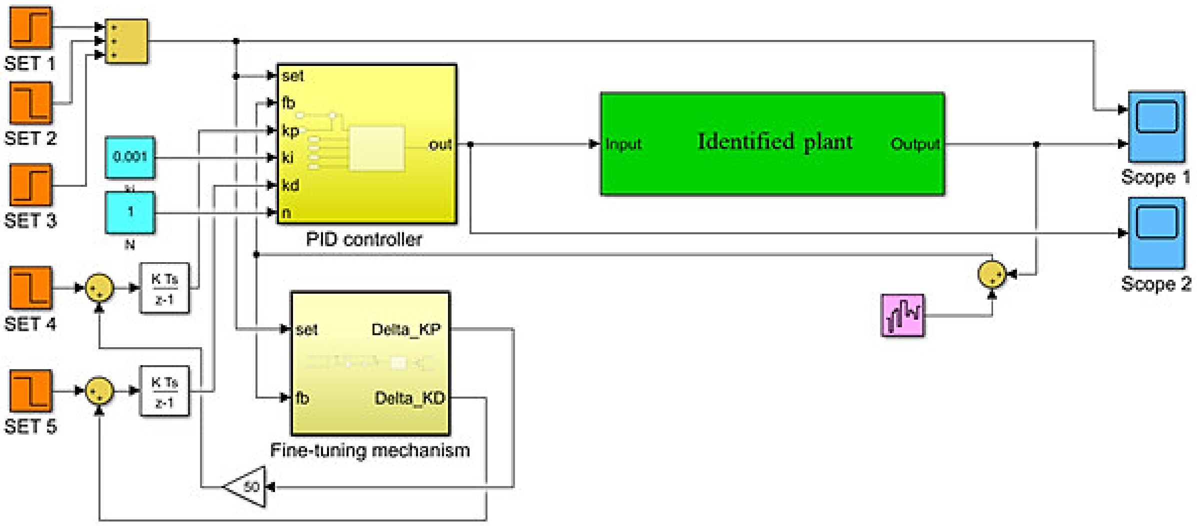
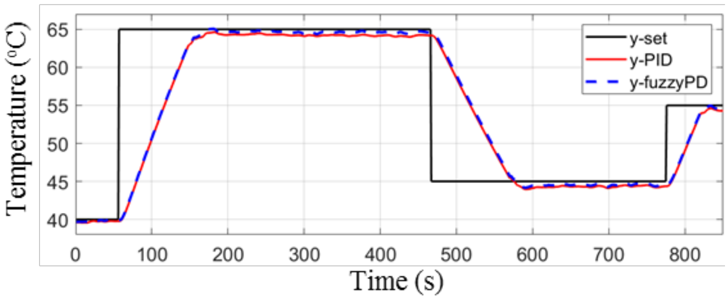
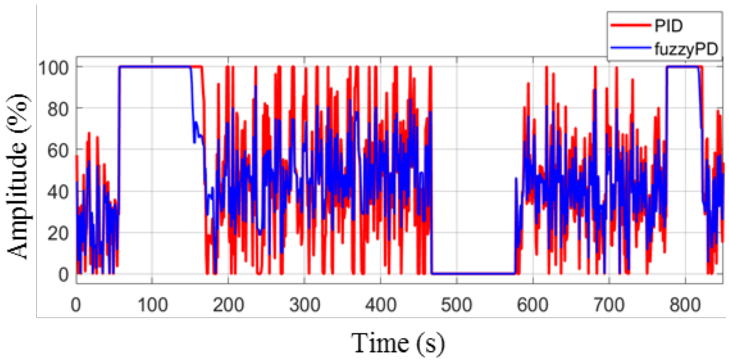


Fig. 7 *The simulation diagram*

Fig. 8 presents the simulation results. The results showed that the fine-tuning algorithm has improved the tracking response of the plant's temperature in terms of the settling error and the control energy. In fact, the settling error was reduced to ±0.5 °C from ±2 °C; the control energy has been reduced and has been more stable than before applying the fine tuning. This proves that K_P and K_D fine-tuned by the fuzzy-PD controller are more optimal than those adjusted and fixed manually. The *Fine-tuning mechanism* block is converted into the SCL module to test the S7-1200 with the real temperature plant.



(a)



(b)

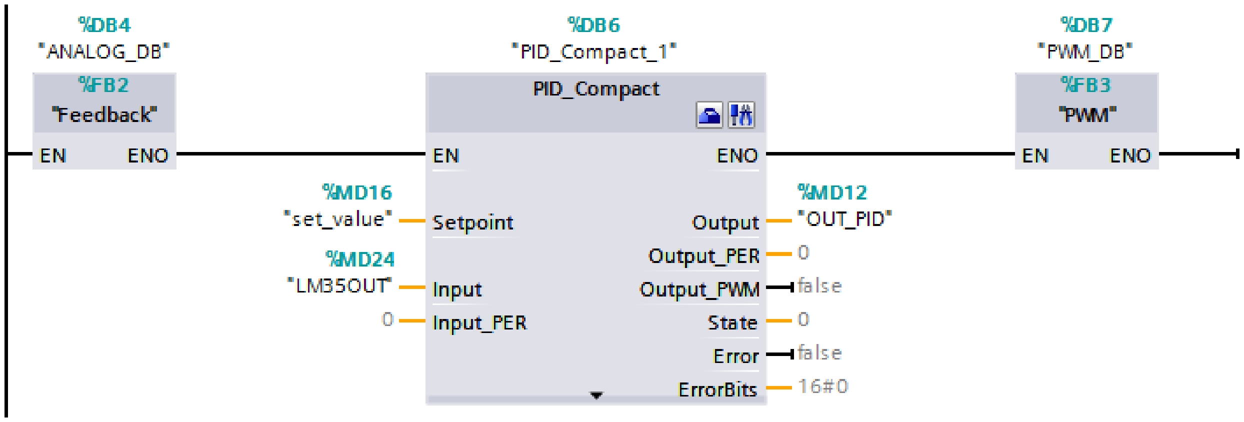
Fig. 8 *Simulation results of the PID controller, with its parameters pre-adjusted by the Ziegler-Nichols method (red solid line) and by the proposed fine-tuning method (blue solid line): (a) temperature response; (b) control law*

3.2.2 Realtime Testing the Fine-Tuning Mechanism on the PLC

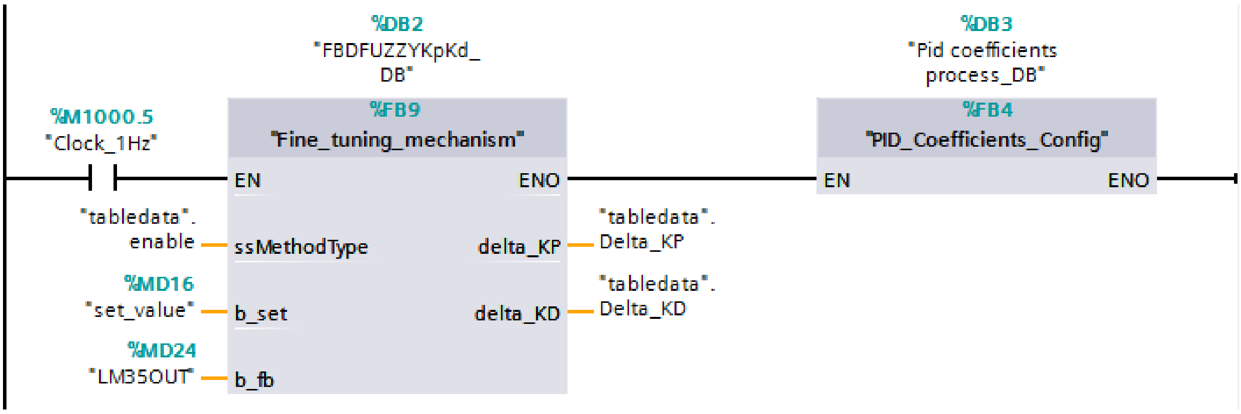
Fig. 9 presents the structure of the testing LAD program of the S7-1200 on the TIA Portal. In the PLC, the built-in PID controller named PID Compact has been used. The PID Compact's parameters have also been adjusted by the Ziegler-Nichols method, as in the simulation. The *Feedback* FB in Fig. 9a reads and filters the feedback temperature from the plant. This value is fed to the input of the *PID_Compact* FB. The *PWM* FB plays a role in converting the output of the *PID_Compact* FB into a PWM signal to change the plant's temperature. The PWM signal was generated by the pulse train output (PTO) of the S7-1200, and its width was mapped to the output of the *PID_Compact* FB. The *Fine_tuning_mechanism* FB in Fig. 9b is the fine-tuning algorithm converted from MATLAB/Simulink. The *PID_Coefficients_Config* FB adjusts the K_P and K_D of the PID Compact by adding them with *Delta_*K_P and *Delta_*K_D, respectively. This FB also constrains the value of K_P and K_D of the PID Compact based on Eqs. (8) and (9), with $K_{P\_min} = K_{D\_min} = 0.00001$ and $K_{P\_max} = K_{D\_max} = 10$. The constraint code shown in Fig. 10 is for the case of K_P, and the same for K_D. The *Fine_tuning_mechanism* FB is activated once every second by the system clock event. The filter, which utilizes Eq. (1), is also implemented using SCL code.

$$K_{P\_\min} \le K_P \le K_{P\_\max} \quad with \quad \begin{cases} K_P = K_{P\_\min} (when \quad K_{P\_\min} \ge K_P) \\ K_P = K_{P\_\max} (when \quad K_P \le K_{P\_\max}) \end{cases} \tag{8}$$

$$K_{D\_\min} \le K_D \le K_{D\_\max} \quad with \quad \begin{cases} K_D = K_{D\_\min} (when \quad K_{D\_\min} \ge K_D) \\ K_D = K_{D\_\max} (when \quad K_D \le K_{D\_\max}) \end{cases} \tag{9}$$



(a)



(b)

Fig. 9 *LAD program: a) the PID compact FB, pre- and post-processing data FB; b) the Fine_tuning_mechanism FB and controller parameters constraint FB*

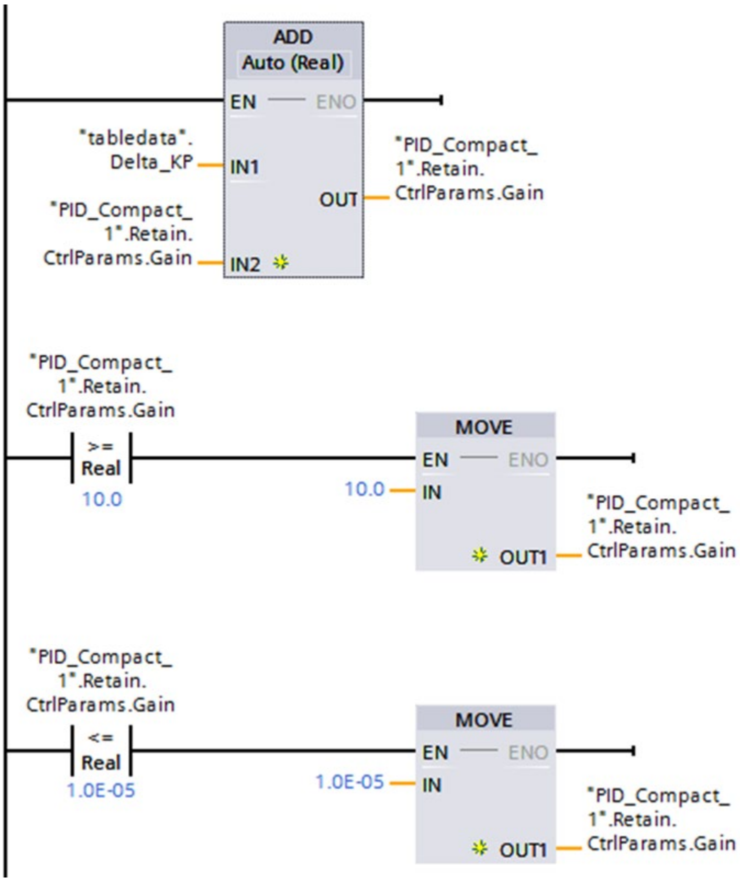


Fig. 10 *Rungs constraining the value of* K_P

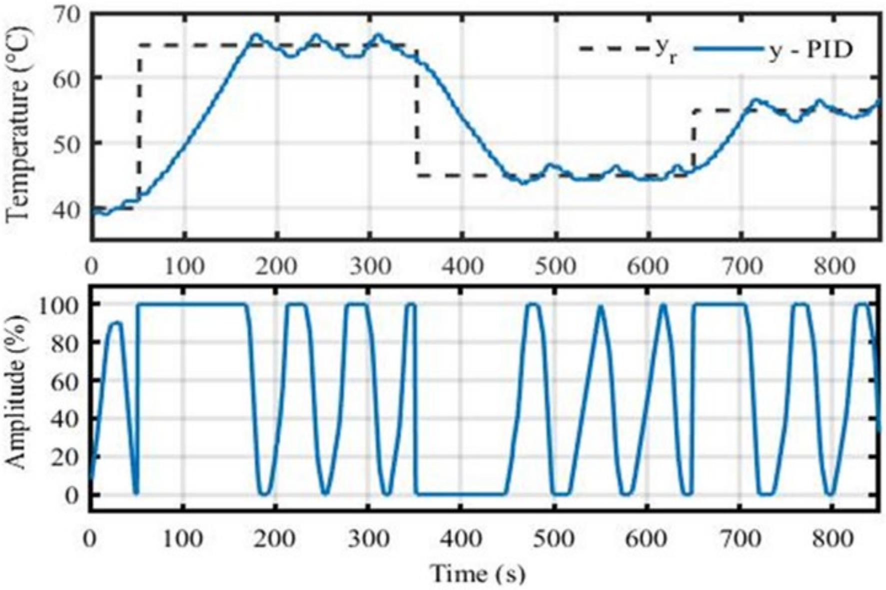


Fig. 11 *Temperature response and control signal on PID compact, with initial parameters*

Fig. 11 illustrates the response of the PID Compact, with the initial parameters adjusted by the Ziegler-Nichols method. As stated above, with the initial parameters, the plant's temperature fluctuates. The role of the Fine_tuning_mechanism FB is to improve the response via fine-tuning K_P and K_D of the PID Compact. Fig. 12 depicts the plant's response in this case. The figure shows that the response is more stable, the overshoot is very low, and the steady-state error is almost eliminated. The adapting processes of K_P and K_D are presented in Fig. 13. Their values are continuously adapting to ensure that the plant tracks the reference well.

Experimental results have proven that the proposed fine-tuning method is effective in adjusting PID parameters. As a result, the automatic system can operate more stably and achieve higher performance. Furthermore, because the PID coefficients are fine-tuned only within a limited range, an unstable system response is avoided. This promises the ability to apply the proposed algorithm to PLCs to solve industrial applications.

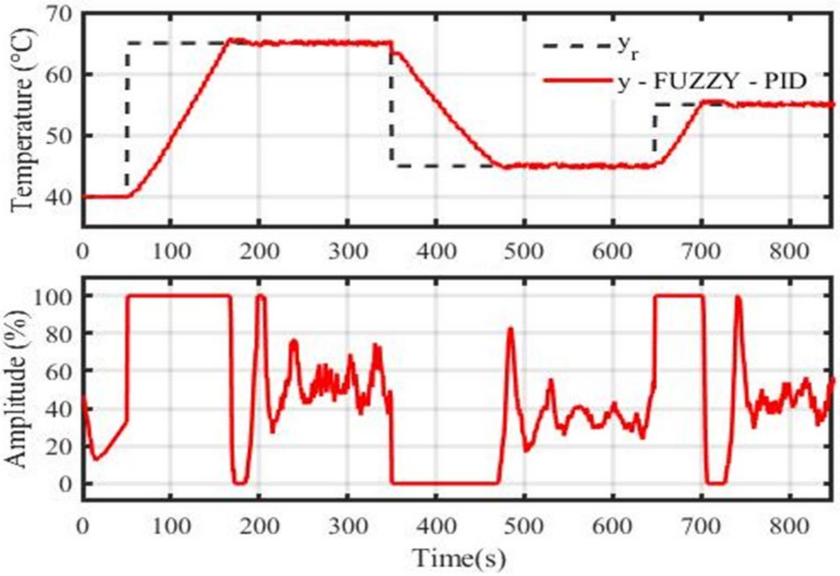


Fig. 12 *Temperature response and control signal of the PID compact, with its parameters fine-tuned by the proposed algorithm*

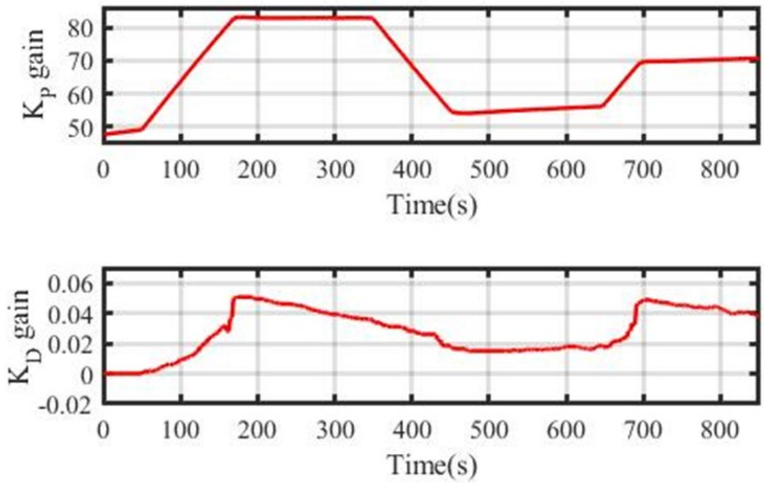


Fig. 13 *The adaptation of K_P and K_D in the control process*

3.2.3 Evaluating the Control Quality

Fig. 14 shows the response of the object when using the PID Compact, with coefficients determined by the built-in Auto-tuning tool. The results showed that the PID Compact controlled the plant's temperature very well. However, the control law in this case is switched at a high frequency (Fig. 14b), which can affect some control plants. Meanwhile, the PID Compact fine-tuned by the proposed algorithm has a stable control law with no chattering phenomenon. In addition, with the proposed algorithm, the PID controller would enable the automatic system to operate stably, especially when the control system's properties change over time.

Table 3 presents the control quality of the PID Compact determined at the time of changing the set temperature from 40 °C to 65 °C. The control quality of the PID Compact is evaluated in two cases, including its parametes pre-tuning by the built-in tool and fine-tuning by the proposed method. The evaluated quality indicators included rise time, overshoot, settling time, and settling error. It can be seen that PID Compact fine-tuning with the proposed algorithm gives a better response than pre-tuning with the built-in tool. Specifically, the overshoot is only 0.79% compared to 0.94%, and the setting error is lower than 0.35 °C.

Table 3 *Control quality of the PID compact with two different auto-tuning methods*

Auto-tuning method	Rise time (sec)	Overshoot (%)	Settling time (sec)	Setting error (℃)
Integrated Auto-tuning tool	54	0.94	113	0.45
Fuzzy-PD-based tuning	58	0.79	118	0.10

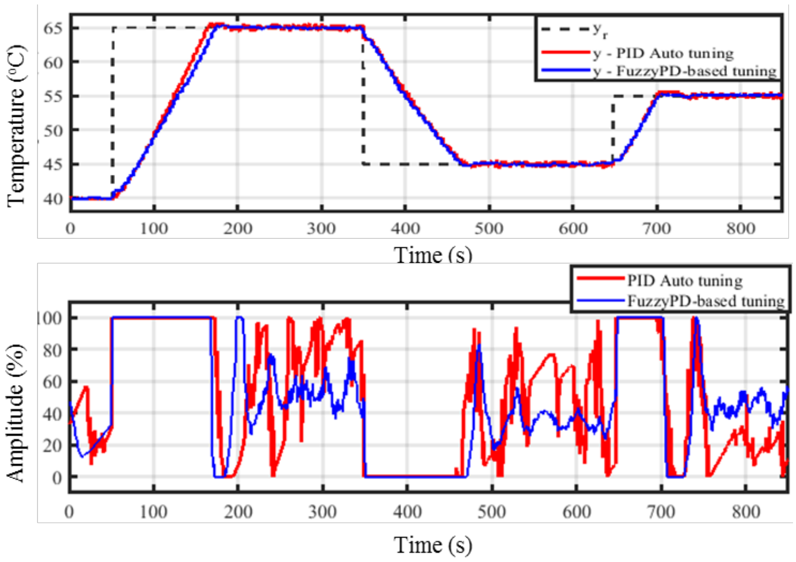


Fig. 14 *Real-time response and control signal of PID compact, with its parameters pre-adjusted by the built-in auto-tuning tool (red solid line) and the proposed fine-tuning method (blue solid line)*

4. Conclusion

This study proposed a method to design and implement a self-tuning PID controller on the PLC using the MATLAB tool, which helps simplify and reduce the time required for PID controller design. The auto-tuning algorithm based on the fuzzy-PD controller significantly improved the system control quality of the PID controller. The experimental results with the Siemens S7-1200 PLC show that the online self-tuning PID controller using the proposed mechanism significantly improved the control response compared with simply tuning once with the integrated tool. Furthermore, the overshoot decreased from 1.94% to 0.79%, and the setting error decreased by 0.35 °C from 0.45 °C to 0.1 °C. These results demonstrate the ability of the proposed method to improve the performance of the PID controller on the proposed method. This helps users to reduce the time required to design a PID controller. In addition, this method can be applied to many other types of PLCs. Similarly, the proposed fine-tuning algorithm also promises applicability to other control plants, instead of just a thermal plant. This study is the fundamental for further research on self-tuning PID algorithms in the future, such as combining neural networks to adjust the parameters of PID controllers.

Acknowledgement

The authors would like to thank Mr. Hoang-Linh Nguyen and Mr. Trong-Huu Mai for supporting controller testing on PLC S7-1200 at the PLC and Industrial IoT lab, Faculty of Automation Technology, College of Engineering, Can Tho University.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design****: Van-Khanh Nguyen, Vy-Khang Tran;* ***data collection****: Vy-Khang Tran, Hai Pham;* ***analysis and interpretation of results****: Van-Khanh Nguyen, Vy-Khang Tran, Hai Pham, Hoang-Dung Nguyen, Chi-Ngon Nguyen;* ***draft manuscript preparation****: Van-Khanh Nguyen, Vy-Khang Tran. All authors reviewed the results and approved the final version of the manuscript.*

Reference

[1] Pastukhov, V. G., & Maydanik, Y. F. (2023). Temperature control of a heat source using a loop heat pipe integrated with a thermoelectric converter, *International Journal of Thermal Sciences*, 184, 108012, https://doi.org/10.1016/j.ijthermalsci.2022.108012

[2] Li, F., Yang, L., Ye, A., Zhao, Z., & Shen, B. (2024). Grouping Neural Network-Based Smith PID Temperature Controller for Multi-Channel Interaction System. *Electronics, 13*(4), 697.

[3] Das, M., Catalkaya, M., Akay, O. E., & Akpinar, E. K. (2023) Impacts of use PID control and artificial intelligence methods for solar air heater energy performance. *Journal of Building Engineering*, 65, 105809.

[4] Shamsuzzoha, M. (Ed.). (2018). *PID control for Industrial processes*. BoD–Books on Demand.

[5] Song, J., Du, J., & Gao, X. (2024). Applying PID control to 2-dof vortex induced vibration of a cylinder. *Ocean Engineering*, 300, 117295.

[6] Awtoniuk, M., Sałat, R., Worwa, M., & Reshetiuk, V. (2024). Implementation of PID Autotuning Procedure Based on Doublet-Pulse Method in PLC Controller. *Advances in Science and Technology Research Journal, 18*(1), 89-97.

[7] Liu, H. (2024). Research on glass intelligent storage control system. International Journal of Frontiers in *Engineering Technology, 6*(1).

[8] Saputra, D., Ma'arif, A., Maghfiroh, H., Chotikunnan, P., & Rahmadhia, S. N. (2023). Design and Application of PLC-based Speed Control for DC Motor Using PID with Identification System and MATLAB Tuner. *International Journal of Robotics & Control Systems, 3(2)*.

[9] Prastiyo, D., & Aji, W. S. (2023). Irrigation Sluice Control System Using Algorithm Based DC Motor PID And Omron PLC. *Control Systems and Optimization Letters, 1(1)*, 19-26.

[10] Zainal, A., Wahab, N. A., & Yusof, M. I. (2023). PLC-based PID controller for real-time pH neutralization process using Palm Oil Mill Effluent. *IIUM Engineering Journal, 24(1)*, 244-255.

[11] Martyanov, A. S., Solomin, E. V., & Korobatov, D. V. (2015). Development of control algorithms in Matlab/Simulink. *Procedia engineering*, 129, 922-926.

[12] Kula, A., Georgieva, A., & Ruzhekov, G. (2022, June). Modelling and Control of Rectification Column. In *2022 22nd International Symposium on Electrical Apparatus and Technologies (SIELA)* (pp. 1-4). IEEE.

[13] Osman, K., Štefić, M., Alajbeg, T., & Perić, M. (2023). Approach in development and implementation of automatic control system in a smart building. *Energy and Buildings*, 113200.

[14] Wei, H. (2023). Solar heat collection photoelectric tracking servo drive system based on cloud computing. *Thermal Science, 27(2 Part A)*, 1141-1149.

[15] Dettori, S., Iannino, V., Colla, V., & Signorini, A. (2018). An adaptive Fuzzy logic-based approach to PID control of steam turbines in solar applications. *Applied Energy*, 227, 655-664.

[16] Algburi, R. N. A., Gao, H., & Al-Huda, Z. (2020). Design and implementation fuzzy-PLC temperature controller for the cooling tower to reduce dust emission in cement plant. In *Developments of Artificial Intelligence Technologies in Computation and Robotics: Proceedings of the 14th International FLINS Conference (FLINS 2020)* (pp. 1270-1279).

[17] Algburi, R. N. A., Gao, H., & Al-Huda, Z. (2020). Design and implementation fuzzy-PLC temperature controller for the cooling tower to reduce dust emission in cement plant. In *Developments of Artificial Intelligence Technologies in Computation and Robotics: Proceedings of the 14th International FLINS Conference (FLINS 2020)* (pp. 1270-1279).

[18] Dequan, S. H. I., Guili, G. A. O., Zhiwei, G. A. O., & Peng, X. (2012). Application of expert fuzzy PID method for temperature control of heating furnace. *Procedia Engineering, 29*, 257-261.

[19] Thuong, T. T., & Ha, V. T. (2023). Adaptive Control For Mobile Robots Based On Inteligent Controller. Journal of Applied Science and Engineering, 27(5), 2481-2487. https://doi.org/10.6180/jase.202405_27(05).0012

[20] Mathworks, Simulink PLC Coder, (2023, October 9). https://www.mathworks.com/products/simulink-plc-coder.html

[21] Yahya, S., Al Tahtawi, A. R., Wijayanto, K., & Faizah, B. (2022). Adaptive fuzzy-pid controller for liquid flow control in the heating tank system. *International Journal of Integrated Engineering, 14(1*), 173-180.

[22] Zhang, H., Zuo, X., Sun, B., Wei, B., Fu, J., & Xiao, X. (2023). Fuzzy-PID-based atmosphere packaging gas distribution system for fresh food. *Applied Sciences, 13(4)*, 2674.

[23] Guzmán, J. L., García-Mañas, F., Hoyo, A., Ramos-Teodoro, J., & Donaire, J. G. (2022). Use of TCLab kits for control engineering curricula at the University of Almería. *IFAC-PapersOnLine, 55(17)*, 362-367.

[24] Ljung, L. (1995). *System identification toolbox: User's guide*. Natick, MA, USA: MathWorks Incorporated.

[25] Smith, S. W. (2003). Moving average filters. Digital Signal Processing, 277-284.

[26] Aras, M. S. M., Basar, M. F., Hasim, N., Kamaruddin, M. N., & Jaafar, H. I. (2013). Development and modeling of water tank system using system identification method. *International Journal of Engineering and Advanced Technology (IJEAT) ISSN*, 2249-8958

[27] Kushwah, M., & Patra, A. (2014). PID controller tuning using Ziegler-Nichols method for speed control of DC motor. *International Journal of Scientific Engineering and Technology Research, 3(13)*, 2924-2929.

[28] Siemen (2019). *Single and Multi Loop Controller Structures (Cascade Control) with PID_Temp*. https://support.industry.siemens.com/cs/ww/en/view/79047707

[29] Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, 7(1), 1-13.