

Concept of the advanced cloud-based autotuning procedure for PID controllers implemented in programmable logic controllers

Michał Frątczak^{1*} , Paweł Nowak¹ , Patryk Grelewicz¹ , Jacek Czczot¹ 

¹ Department of Automatic Control and Robotics, Silesian university of Technology, Akademicka 16, 44-100 Gliwice, Poland

* Corresponding author's e-mail: michal.fratczak@polsl.pl

ABSTRACT

This paper presents a concept of the cloud-based tool that extends the auto-tuning capabilities accessible in the PID function blocks available in programmable logic controllers. On the basis of the data collected during the built-in auto-tuning experiment, a dynamical model of the process was obtained in the cloud environment and then by any more or less advanced method a new set of PID tunings was proposed based on different closed-loop performance criteria predefined by the user. This method can be an alternative for the for the autotuning methods accessible in the PID function blocks and in this paper, its usability in industrial control systems was validated using virtual commissioning technique based on Siemens equipment and programming environments.

Keywords: auto-tuning method, PID controller, virtual commissioning, cloud-based implementation.

INTRODUCTION

Nowadays, one of the most important challenges in industrial automations is to improve production quality and, at the same time, reduce production costs by, e.g. optimization of energy consumption. This aspect is one of the most important priorities of the European Union [1]. Potentially, reduction of energy consumption can be achieved by upgrading the control equipment, which is always associated with high costs. However, very often such a modernization does not bring the expected results, because of inefficient control algorithms that are still implemented in the control equipment, even after the upgrade. This is confirmed by the reports conducted by professionals from the European Union, which identify a common cause of inefficient production as poorly configured control algorithms. When this problem concerns the PID controllers which still are the most common control algorithm implemented in industrial control systems,

the greatest problem results mainly from their incorrect tuning.

Proper tuning of industrial PID controllers require a lot of and experience, which is not very common among the industrial automation engineers. Additionally, many processes are time-varying and/or nonlinear, which requires modifying the controller tunings depending on the current operating conditions. Additionally, nowadays, the conditions determining the required product quality change frequently which is also a great challenge for the operation of PID controllers, because this operation should be adapted to the current product requirements. These problems are critical for the production efficiency and thus the as well as and reliable auto-tuning methods for PID controllers are an effective solution to these problems. The first auto-tuning methods were proposed in the 1980s [2]. Currently, as this auto-tuning functionality is highly desirable in the practice, it is available in almost every PLC-based implementation of PID algorithm. Shortly speaking, the auto-tuning procedure is the

on-demand functionality that does not require any additional expert knowledge about tuning PID controllers. However, the auto-tuning is not significantly different from the procedure of controller tuning carried out by an experienced engineer. In the first step, a kind of an excitation signal (in a form of step change or square waveform) should be applied to the process control input and the process response data should be collected. On the basis of these data, a simple model is usually identified and in the majority of cases it has the form of first order plus dead time (FOPDT) dynamical model. Finally, controller tunings are somehow calculated based on the identified model parameters. Many different auto-tuning algorithms can be found in the literature, e.g. [3–5] and also, all leading PLC manufacturers have their own implementations. However, the practical experiences show that in the case of some processes with more complex dynamics or nonlinear characteristics, the PID tunings obtained based on the build-in auto-tuning methods do not ensure satisfactory control performance and further manual retuning of the controller is required. This fact stands as the most important motivation for the works presented in this paper.

New technologies are developing rapidly in modern world. One of them is cloud computing, which is one of the pillars of Industry 4.0. Cloud computing is based on the accessibility to computing services performed externally [6] or implemented locally on the specialized equipment that ensure much higher computational computing power comparing to even modern PLC platforms. The PLCs are still the most popular platform for implementing PID controllers in the industrial control systems, mainly due to the accessibility of dedicated PID function blocks that make implementation fast and easy. With edge, cloud or fog computing, it is possible to extend the functionalities of PLCs without significant interferences with the existing control program. The example of such a cloud computing-based system that can be used in the industrial control systems is the control performance assessment system presented in [7]. The other example is the system which is used to control a part of robotic arm controlled by PLC supported by cloud computing [8].

Similarly to the development of the hardware and software industrial platforms, there is also a continuous progress in developing new and more effective PID tuning methods. In general, the PID tuning methods can be divided into two

groups [9]: classical and intelligent tuning methods. The classical methods have been known for decades and they are still developing. The latest works in this area can be found in literature, e.g. [10] where the authors proposed the experimental PID tuning method based on the selected samples of the sampled process step response. They use symmetric send-on-delta sampling method to avoid closed loop oscillations. This method is further extended for optimal tuning of PID controllers with derivative filter [11] and validated by simulation and experimentally using double-tank laboratory setup. The extension of the classical PID tuning method based on the simplified SOPDT local process approximation is presented in [12] where the authors proposed how to use the extended linearization method to derive the nonlinear SOPDT model and how to use this model for PID tuning for nonlinear processes.

The second group of intelligent tuning methods have much higher computational complexity but due to increase of the computational power of the modern control equipment, they become more and more attractive for practitioners, because potentially they can be implemented directly in the industrial control systems. The optimization-based methods surely belong to this group and the selected examples of the latest reports on this subject can be found in the literature. In [13] it was shown how to tune PID controller optimally to ensure energy efficiency of the controlled process. The authors proposed their method for multivariable systems and present its efficiency by simulation for the control of the air handling unit. The current popularity of machine learning methods is also reflected in the creation of new advanced PID tuning methods based on optimization. For example, in [14] it was proposed how to use the fuzzy-based tuning approach dedicated to precise tuning of the conventional PID controller implemented in PLC. The application of the reinforcement learning for tuning of PID controllers dedicated to control nonlinear processes is presented in [15] where the authors proposed using the twin delayed deep deterministic policy gradient algorithm and show superiority of their approach over the conventional tuning methods in the application to the biological dynamic system. In [16] the authors proposed PID tuning method based on the mixed adaptive artificial bee colony and fuzzy logic approach. This method was used for tuning PID controller operating in automatic voltage regulator system and implemented in

a Siemens S7-1200. Significant progress is also noticeable in the development of PID autotuning methods. In [17], the authors proposed a modified continuous-cycling method for the efficient PID autotuning. The gain of the proportional controller is updated automatically and this method allows obtaining the accurate process frequency response even in the presence of the disturbances. Autotuning method for the filtered PID controller was proposed in [18] for the processes operating under certain operating conditions. The method is based on optimization approach using H_2 norm and its efficiency is shown experimentally. The PID autotuning method dedicated for control of robot manipulators can be found in [19] and it is based on the modified relay feedback test with optimization under the uncertainty principle. In [20], the authors proposed another improvement of the PID autotuning functionality based on the FOPDT process approximation and optimization method to minimize the integral time absolute error with the target phase margin of the closed loop system. PID autotuning procedure implemented in PLC Siemens S7-1200 is reported in [21] where the authors proposed the double-pulse identification of the dynamics of the controlled process in the form of the FOPDT approximation and then use this model for PID tuning. Dedicated software tools are also developed and proposed for practical use. The example of such tools can be found e.g. in [22] where PID tunings were calculated by sub-optimal compromise between robustness, performance and control effort of the closed loop system. Another example can be found in [23] where the authors described Rex-Tuner – a software tool for tuning PID controllers implemented in Siemens family PLCs using the function blocks PID-Compact, PID-3step and PIDF-Temp. This work describes the tool itself and the method how it should be integrated with the existing PLC-based control system. The readers interested in a more detailed overview of the PID tuning methods are referred to many review publications, e.g. [9], [24] and for more complex nonlinear PID controllers [25].

This paper proposed a system that is based on the operation of the auto-tuning procedure built in a Siemens S7-1500 PID Compact function block using the results of this procedure as the reference point. The concept directly uses the experiments that are carried out during the built-in auto-tuning functionalities and it is implemented using cloud computing architecture. The

data collected during the built-in auto-tuning experiments is used to create a dedicated dynamical model of the controlled process and then this model is used for calculating the PI controller settings using a few selected methods including the standard FOPDT-based PID tuning methods and the dedicated optimization-based method. The process operator can compare the closed-loop performance for all the considered methods, select the one that is optimal in the current conditions and initiate bumpless switching between the current (ineffective) PID settings to the tunings suggested by the proposed cloud-based system. The results proving the efficiency of the proposed system are obtained using the virtual commissioning approach in which the real system is represented by the dynamical system with high and time-varying dynamics simulated in the industrial Siemens SIMIT environment.

STATEMENT OF THE PROBLEM

The purpose of this work was to propose a system that allows improving the performance of the control loop in comparison with the built-in auto-tuning functionality. In addition, this system should meet the following requirements:

- from the user viewpoint, the complexity of the system functionality should be comparable with built-in auto-tuning procedure. Namely, it should be user-friendly for non-experienced process operators;
- its performance should not disrupt the normal operation of the control system excluding the need to conduct an experiment by the built-in auto-tuning procedure started by the user demand.

Both of the aforementioned requirements can be satisfied by using cloud computing technique for the implementation. The procedure for determining new (improved) tunings is proposed as follows:

- a) After starting the auto-tuning experiment built in the PLC-based PID controller function block, the manipulated and output process data is collected and sent to the cloud resources. Then, based on collected data, the dynamic model parameters are identified. In the general case, the model can have different forms, e.g. FOPDT or second order plus delay time (SOPDT). It is commonly known that a more accurate model leads to obtaining better set of tunings for PID

controller. Nevertheless, even based on the simple FOPDT model, it is possible to improve the performance of the control loop by simulation testing a wider range of potential tuning methods comparing to those proposed by the auto-tuning method. This approach is compatible with virtual commissioning approach [26] which is another important pillar of Industry 4.0.

- b) In general, depending on structure of obtained dynamic model, various tunings can be generated and their performance can be initially verified by simulations. In the case of FOPDT or SOPDT dynamic models, PID tunings can be calculated based on well-known deterministic tuning methods [27]. In the case of more complex models, general-purpose tuning methods are not available and thus, numerical optimization-based methods can be used to generate effective tunings by optimizing selected performance indices.
- c) The process operator obtains a tuning report from cloud-based system containing several sets of tunings with their impact on the control performance. Finally, a PID function block can be bumplessly switched to a new set of tunings or its operation can be continued with tunings obtained from the built-in auto-tuning procedure.
- d) This procedure executes automatically after initialization of auto-tuning functionality built in PLC-based PID function block.
- e) The proposed procedure is general and it can be applied for a wide range of processes after

initial adjusting the structure of the approximating dynamic model to the dynamics of the real process.

ARCHITECTURE OF THE PROPOSED CLOUD BASED SYSTEM

The example general architecture for the proposed cloud based auto-tuning system is presented in Figure 1 and it can be divided into three logical modules. The first module is a MQTT Broker based on Eclipse Mosquitto that is responsible for communication between PLC and the other modules of the cloud based system. Communication is fully secured because of the ability to use the latest hashing algorithms, such as SHA256. In addition, it is possible to handle users and passwords, and support certificates using asymmetric keys. The data collected during PLC-based PID auto-tuning procedure, jointly with the other measurement signals that can be necessary and are accessible in PLC and the status of auto-tuning procedure is published by PLC to the MQTT Broker module and then, it is stored in Influx time series database. Analytic module is responsible for identification of process model parameters based on selected data accessible in Influx and computing new sets of PID tunings. Finally, calculated tunings are published to the MQTT Broker module which are also subscribed by PLC MQTT Client.

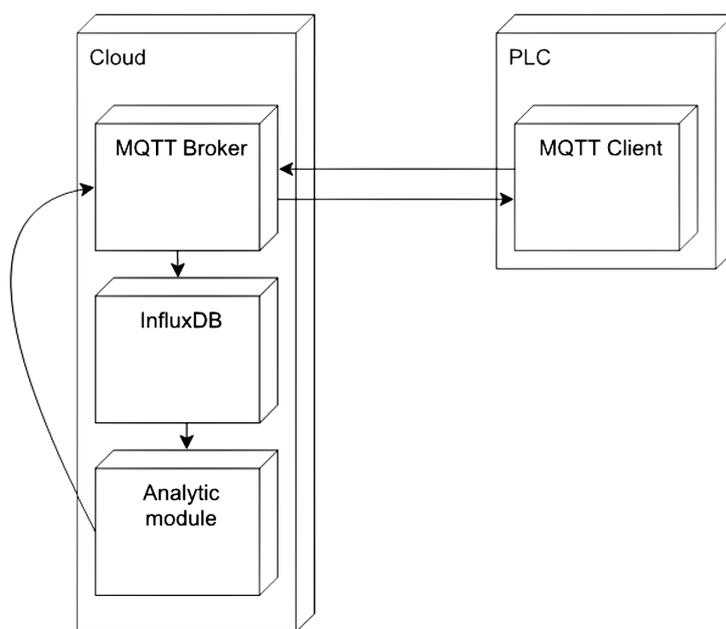


Figure 1. Schematic diagram of cloud modules and their communication with PLC

The MQTT_Client block implemented in PLC is responsible for communication with the broker and it enables publishing and subscribing data. Published and subscribed data is serialized and deserialized, respectively, and it should have well-defined structure for proper decoding variables. Figure 2 shows the implementation of MQTT_Client block with prepareDataToSend function which allows serializing respective variables before publishing.

In this work, PLC implementation is based on the Siemens environment that includes Siemens S7-1500 unit in which the PID Compact function block is implemented. Thus, the auto-tuning procedures (pretuning and finetuning) built in this block are used for collecting the data. The dynamics of the process in this system are approximated using the FOPDT model. The parameters of this model are determined by optimizing the quality index, which is the sum of the squares of the error between the process response and the model response. The Nelder-Mead algorithm [28] was used for this purpose.

The proposed solution shows the potential application of cloud-based system to support PLC calculations. Considering limited PLC resources, application of cloud-based resources enables efficient identification of model parameters, even for large model complexity. Easy access to the data stored in cloud based system is also very convenient when it is required to provide the connection to a predesigned user interface, e.g. implemented in a web browser.

MODEL OF ELECTRIC FLOW HEATER

The concept proposed in this paper was validated using virtual commissioning technique that requires that the process to be controlled be simulated in the dedicated programming environment. For this purpose, the model of the electric flow heater used is applied to create a digital twin of the real process. In order to preserve possibly highest level of compliance with the real process, this model consists of two parts. One part directly describes its physical nature and is based on the heat conservation principle. The second part describes the additional dynamics that potentially exists in the real system, but is not included in the physical part.

The schematic diagram of the considered electric flow heater is presented in Figure 3. Assuming its perfect insulation and water as the flowing and heated medium, the physical part of the model is given by the following equations:

$$\frac{dT_{out}^*}{dt} = \frac{F}{60V} (T_{in} - T_{out}^*) + \frac{P_h P_{nom}}{100 c_s \rho_s V} \quad (1)$$

where: $c_s = 4200 \text{ J/(kg } ^\circ\text{C)}$ is a water specific heat, $\rho_s = 1 \text{ (kg/L)}$ is water density and $V = 1.6 \text{ (L)}$ is the volume of the heater, $P_{nom} = 12 \text{ (kW)}$ is the nominal power of the heater and P_h (operated within 0 and 100%) is the percentage of the nominal power P_{nom} .

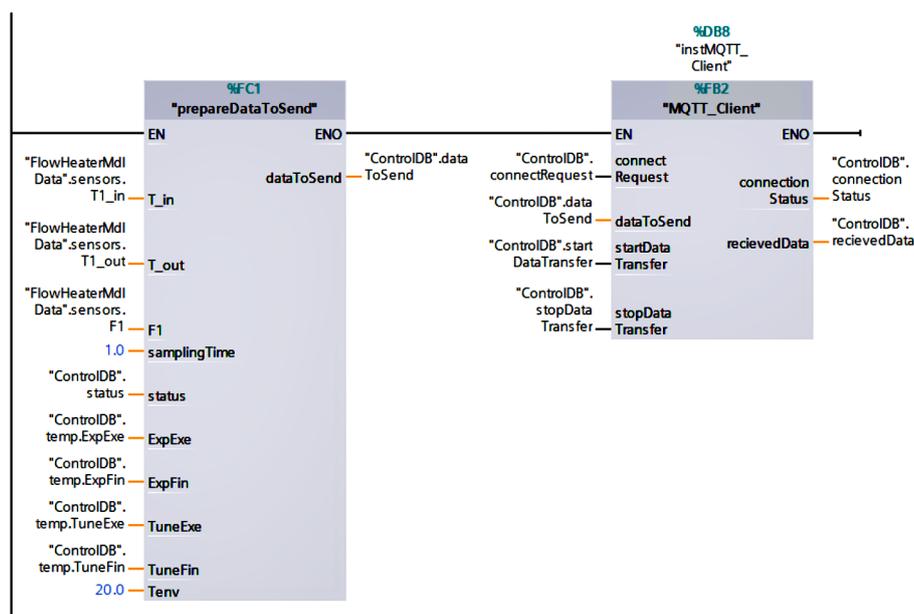


Figure 2. Program implementation in PLC for communication with cloud-based resources

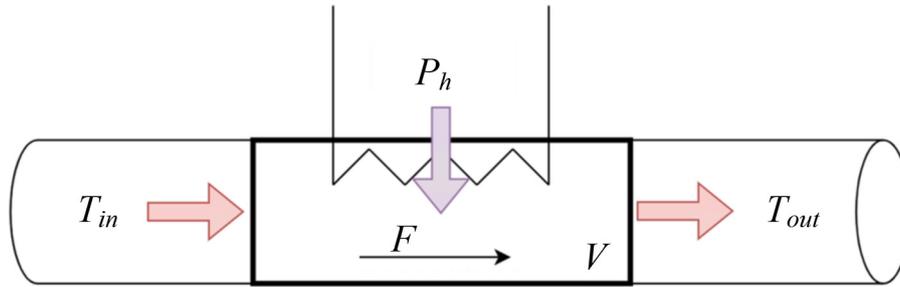


Figure 3. Schematic diagram of an industrial electric water heater used for deriving its digital twin

F (L/min) represents the flow rate of the water, T_{in} ($^{\circ}\text{C}$) denotes the measurable inlet temperature, while T_{out}^* ($^{\circ}\text{C}$) is the unmeasurable (internal) temperature that stands as the output of the first principle dynamical Equation 1.

Temperature T_{out}^* also stands as the input to the dynamics represented by the transfer function (2) and represented the physical phenomena not described by Equation 1:

$$K(s) = \frac{T_{out}(s)}{T_{out}^*(s)} = \frac{k(P_h)}{(1+s\tau_1(F))(1+s\tau_2(F))} e^{-s\tau_0(F)} \quad (2)$$

where: T_{out} ($^{\circ}\text{C}$) denotes the temperature at the outlet of the electric flow heater, k is the gain that potentially depends on the power P_h , time constants τ_1 , τ_2 represent additional dynamics potentially depending on the flow rate F and τ_0 is the dead time the value of which can also depend on the flow rate F .

The proposed relationships that describe variations of the parameters of the additional dynamics (2) according to the changes of the operating point defined by the flow rate F have been identified based on real measurement data and are expressed as:

$$\begin{aligned} k(P_h) &= -0.0002347 \cdot P_h + 1.012 \\ \tau_1(F) = \tau_2(F) &= 19.08 \cdot F^{-0.4293} - 4.042 \quad (3) \\ \tau_0(F) &= 11.93 \cdot F^{-0.78} - 2.37 \end{aligned}$$

Note that the model (1-3) has complex third order dynamics with dead time with the non-linearity resulting from time-varying parameter of the model dynamics. From the control point of view, the input to this process is the percentage power P_h and the output is the temperature T_{out} .

VIRTUAL COMMISSION AND VALIDATION TESTS

The validation of the proposed concept of the cloud-based advanced autotuning system was carried out using the virtual commission technique and it was based on the Siemens equipment and programming environment. The model of the real process presented in section 4 was implemented in the Siemens SIMIT environment. This environment is dedicated for conducting virtual commissioning of the industrial control systems. This implementation is shown in Figure 4.

The Siemens SIMIT environment with implemented model was connected to a Siemens S7-1516 controller running in the PLC Advanced program. A control system consisting of a PID Compact block available in the TIA Portal software was designed for this PLC controller. Two auto-tuning procedures are implemented in the Siemens PID Compact function block: pre-tuning and fine tuning. Both of them can be used to obtain data for model identification, but this paper concentrates on the pre-tuning procedure. For the flow rate assumed as $F = 2$ (L/min), the dynamics of the simulated process is approximated using the following FOPDT model:

$$K_{FOPDT}(s) = \frac{T_{out}(s)}{P_h(s)} = \frac{0.86}{56.1s+1} e^{-26.6} \quad (4)$$

and Figure 5 shows how the FOPDT approximation (4) matches the simulated process model (1)-(3) in the pre-tuning experiment. Note that FOPDT accurately represents the dynamics of the simulated process with the exception of the additional dead time introduced by FOPDT approximation due to its simplified form.

For FOPDT process approximation (4), different tunings methods can be considered. Some of them can be called conventional and are easily accessible in the literature. In this paper, the methods such as Chien-Hrones-Reswick method, 0%

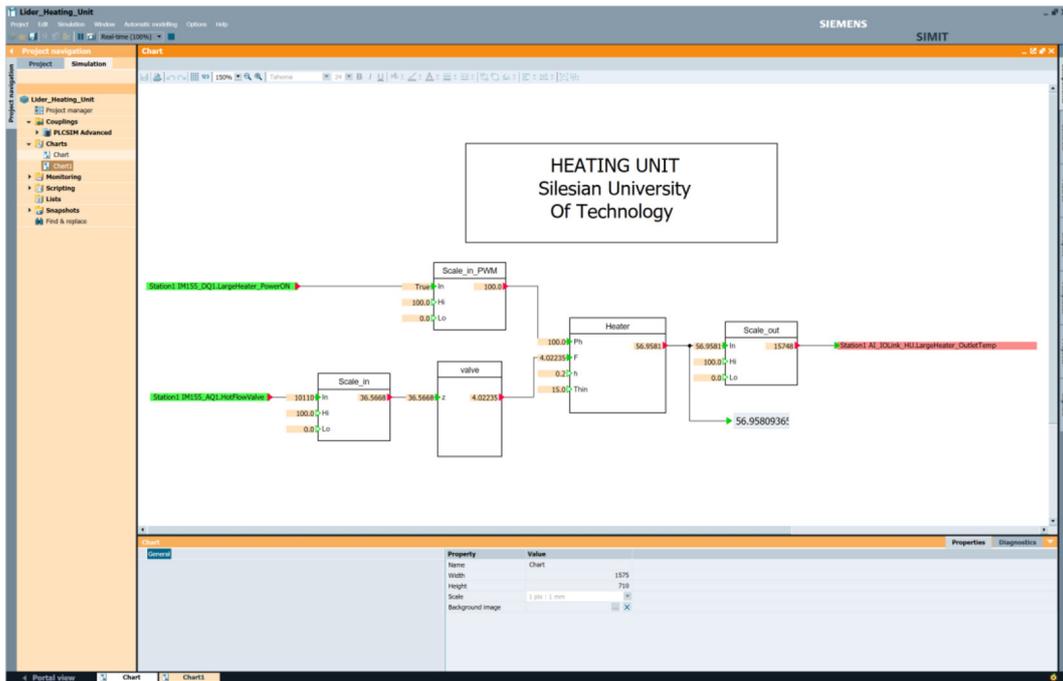


Figure 4. Flow water heater model implemented in the Siemens SIMATIC environment

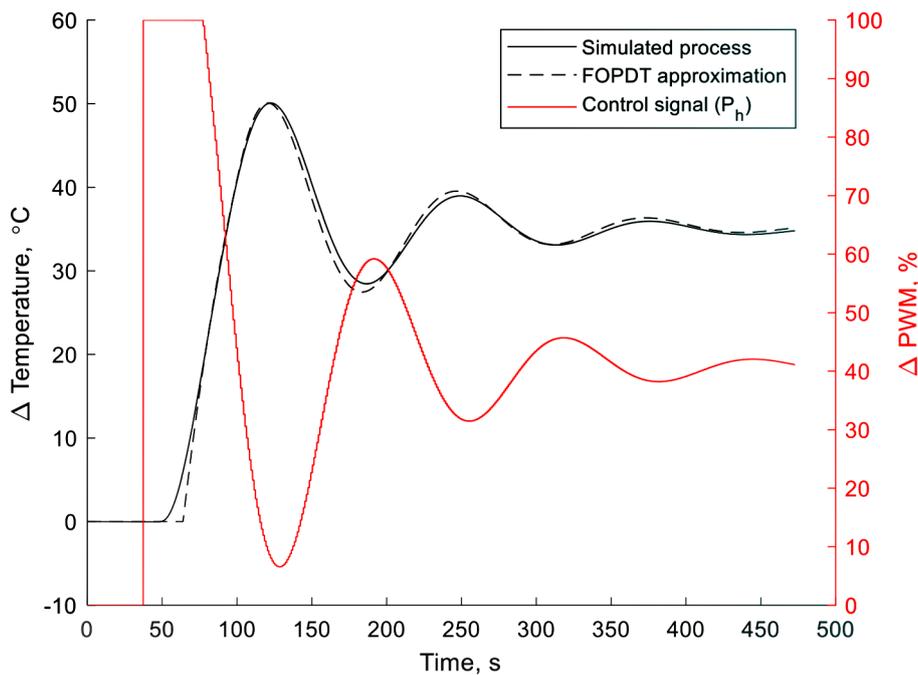


Figure 5. Verification of the accuracy of the flow water heater model

overshoot servo mode (here denoted as CHR0), and the Chien-Hrones-Reswick method, 20% overshoot servo mode (here denoted as CHR20) were selected for validation [29]. This choice is justified by the fact that in the documentation of the PID compact function block, one can find information that the Chien-Hrones-Reswick method is used in the autotuning procedure. It should be

emphasized, however, that in literature there are several methods referred to as Chien-Hrones-Reswick. They differ in the declared overshoot and in the adjustments to follow setpoint changes or disturbance suppression. The PID Compact function block documentation lacks detailed information on which variant of the Chien-Hrones-Reswick method was selected. Additionally, SIMC method

[30] and the AMIGO [31] method were also used for comparison. They are popular tuning methods, often chosen as benchmarks when testing various algorithms and tuning rules. Examples of new works using these methods can be found in [32–34]. On the basis of these methods, the gain of controller PID k_r and its integration time constant T_i were calculated using respective formulas based on the parameters of the FOPDT process approximation (4). For the closed-loop experiment using the step change of the setpoint as the excitation, the results for the considered tuning methods are shown in Figure 6. For clarity, they were compared with the PID settings obtained directly from pre-tuning and fine-tuning methods built in PID Compact function block. Table 1

additionally shows the values of the selected control performance indicators:

- maximum overshoot

$$maxOvershoot = 100 \frac{\max(y(t)) - y(t_{final})}{y(t_{final})} \quad (5)$$

- integral absolute error

$$IAE = \int_0^{t_{final}} |e(\tau)| d\tau \quad (6)$$

- integral time absolute error

$$ITAE = \int_0^{t_{final}} \tau |e(\tau)| d\tau \quad (7)$$

- settling time, here calculated as the time after which $|e(t)| < 0.02y_{sp}$.

where: Y is the process output, t_{final} is the time of stabilization of the process output, e is the control error and finally, y_{sp} is the setpoint.

Table 1. Control performance indices for different tuning methods (literature review)

| Tunings | k_r | T_i | Indices | | | |
|-------------|-------|-------|---------------|-------------|-------------|---------------|
| | | | Max overshoot | IAE | ITAE | Settling time |
| CHR0 | 0.86 | 65.7 | 0.08 | 88.7 | 6403 | 310.2 |
| CHR20 | 1.47 | 56.1 | 11.65 | 55.9 | 2166 | 152.1 |
| SIMC | 1.23 | 56.1 | 4.05 | 57.6 | 2085 | 161.9 |
| AMIGO | 0.5 | 60.5 | 0.79 | 140.2 | 16237 | 487.7 |
| Pre tuning | 2.69 | 87 | 19.53 | 66.7 | 4904 | 368.6 |
| Fine tuning | 2.54 | 77 | 19.81 | 63.57 | 4150 | 283.6 |

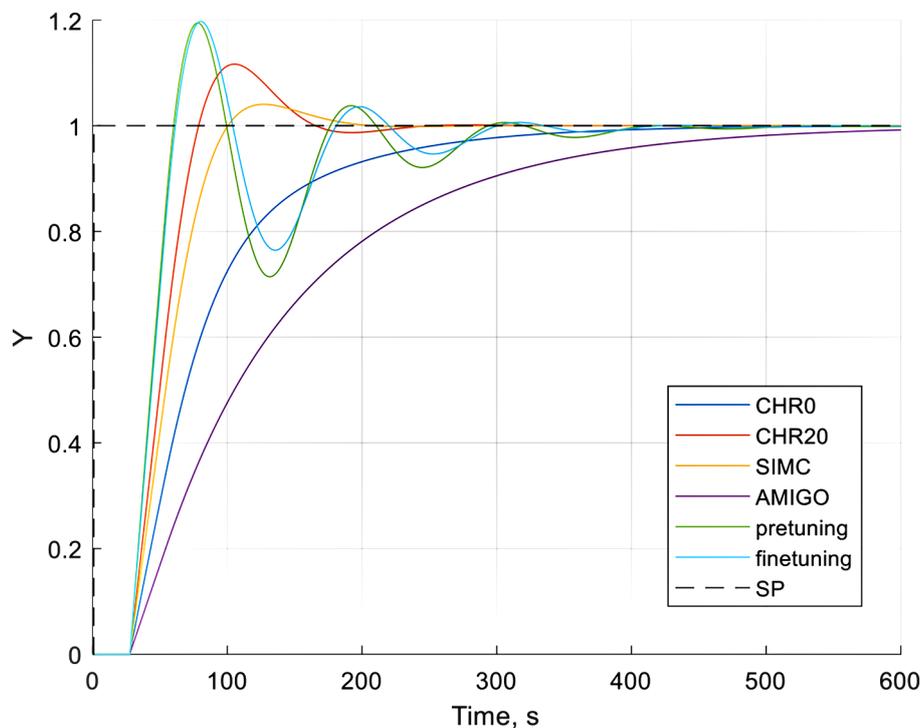


Figure 6. Responses of obtained dynamical model for a step change of the setpoint for PI tunings based on the literature review

In the second stage, a series of PI controller settings were determined numerically in the cloud-based environment to directly minimize the control indices (5)–(7) defined above. In this paper, the following nomenclature is used: tunings A denote minimization of overshoot (5), tunings B denote minimization of IAE (6), tunings C denote minimization of ITAE (7) and finally tunings D denote minimization of settling time.

The Nelder-Mead algorithm [28] was used to find the sets of PI tunings, which minimize above indices for the closed loop performance in the response of the step change of setpoint y_{sp} and Table 2 shows the obtained results. Minimal values of the considered indices are bolded. Once again, the tunings directly determined by the PID

Siemens built-in auto-tuning methods (pre-tuning and fine-tuning) are included with the corresponding values of the considered indices. Figure 7 shows responses of the closed loop performance obtained in the presence of the step change of the setpoint for computed PI tunings. On the basis of the simulated time responses shown in Figures 6 and 7 and the data shown in Tables 1 and 2, the process operator can make a final decision about the best set of tunings and thus, retune the PI controller. From the practical viewpoint, this completes the auto-tuning procedure. The proposed methodology was verified in cloud-based environment described in section 3 using virtual commissioning method with the digital twin (1)-(3) representing the real process. PI control algorithm

Table 2. Control performance indices for different tuning methods

| Tunings | k_r | T_i | Indices | | | |
|-------------|-------|-------|---------------|-------------|-------------|---------------|
| | | | Max overshoot | IAE | ITAE | Settling time |
| A | 0.52 | 50.9 | 0 | 114.0 | 9504 | 327.8 |
| B | 1.6 | 65.2 | 9.3 | 54.2 | 2162 | 202.1 |
| C | 1.34 | 59.3 | 4.85 | 55.6 | 1998 | 145.2 |
| D | 1.34 | 63 | 1.99 | 56.0 | 2113 | 93.1 |
| Pre tuning | 2.69 | 87 | 19.53 | 66.7 | 4904 | 368.6 |
| Fine tuning | 2.54 | 77 | 19.81 | 63.57 | 4150 | 283.6 |

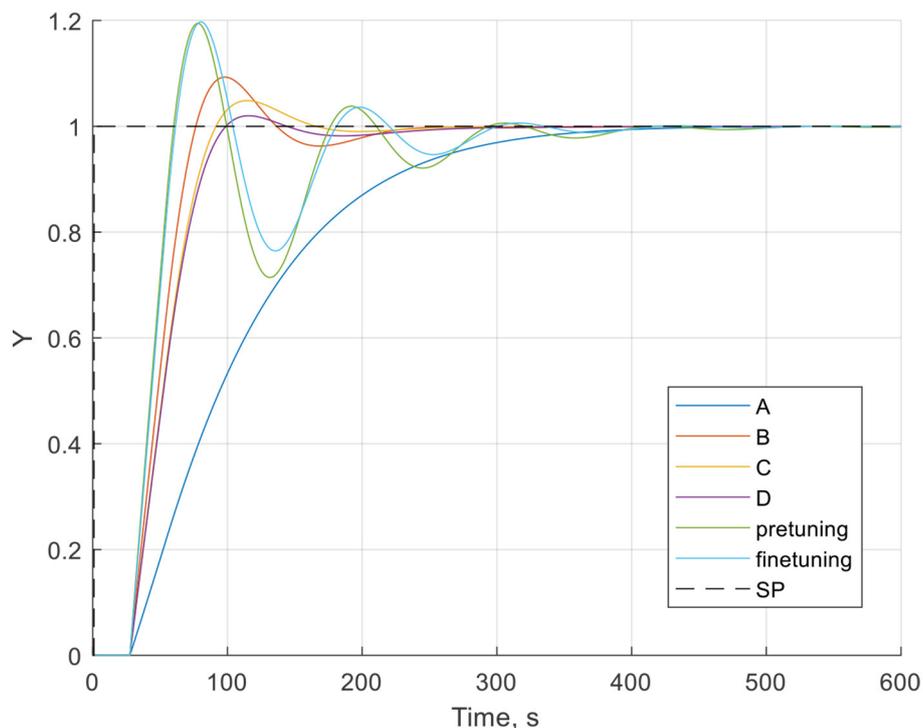


Figure 7. Responses of obtained dynamical model for a step change of the setpoint for PI tunings based on minimization of selected indices

implemented as PID Compact function block was tuned based on optimization-based tuning method D and the conventional SIMC method. They both were arbitrarily selected as the best representatives of each group considered above.

The verification experiment consisted of two stages. Firstly, the setpoint was changed from 35 °C to 50 °C. After 600 seconds, the setpoint was again changed from 50 °C to 40 °C. Figures 8 and 9 show a comparison between the closed loop performance obtained for the selected tunings

methods and for the pretuning as well fine tuning methods built in the PID Compact function block. In the results shown in Figure 8, the medium flow was set as $F = 2$ (L/min), which represents the operating point for which the FOPDT approximation (4) used for PI tuning was determined. In turn, the results shown in Figure 9 were obtained for the medium flow $F = 1.5$ (L/min), which represents the operating point that is different from the one used for determining PI tunings. Readers should notice a deterioration in control quality at reduced

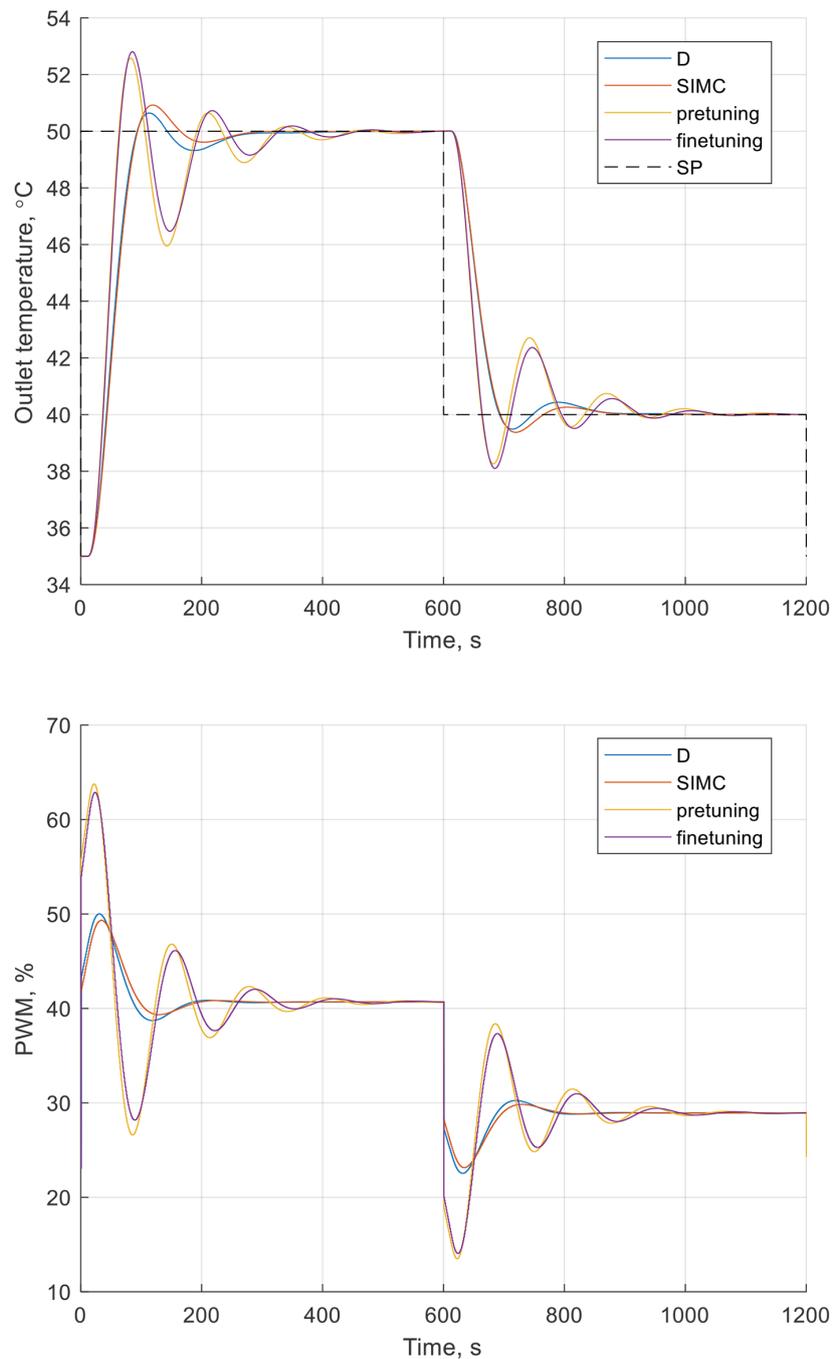


Figure 8. System response and control signal for change of setpoint. Flow set at 2 l/min

flow because no retuning was applied here. Application of pretuning and finetuning-based tuning parameters leads to closed-loop oscillations, while the methods used in the cloud-based system still provide acceptable control quality.

Tables 3–6 present the calculated values of the control indices separately for each setpoint change and for different flow rates. The numerical values of these indices confirm the observations drawn based on the graphs shown in Figures 8 and 9. The pretuning and finetuning methods in

this case differ significantly from the SIMC tuning rule and the selected D optimization based tuning method. In each case, D tuning method ensures the shortest settling time and the best (or very close to the best) values of the other considered indicators. It is worth noting that the finetuning method ensures slightly better results than the pretuning method. This is consistent with the manufacturer’s assumptions for the autotuning functionalities built in the PIDCompact function block. The differences between the SIMC and D

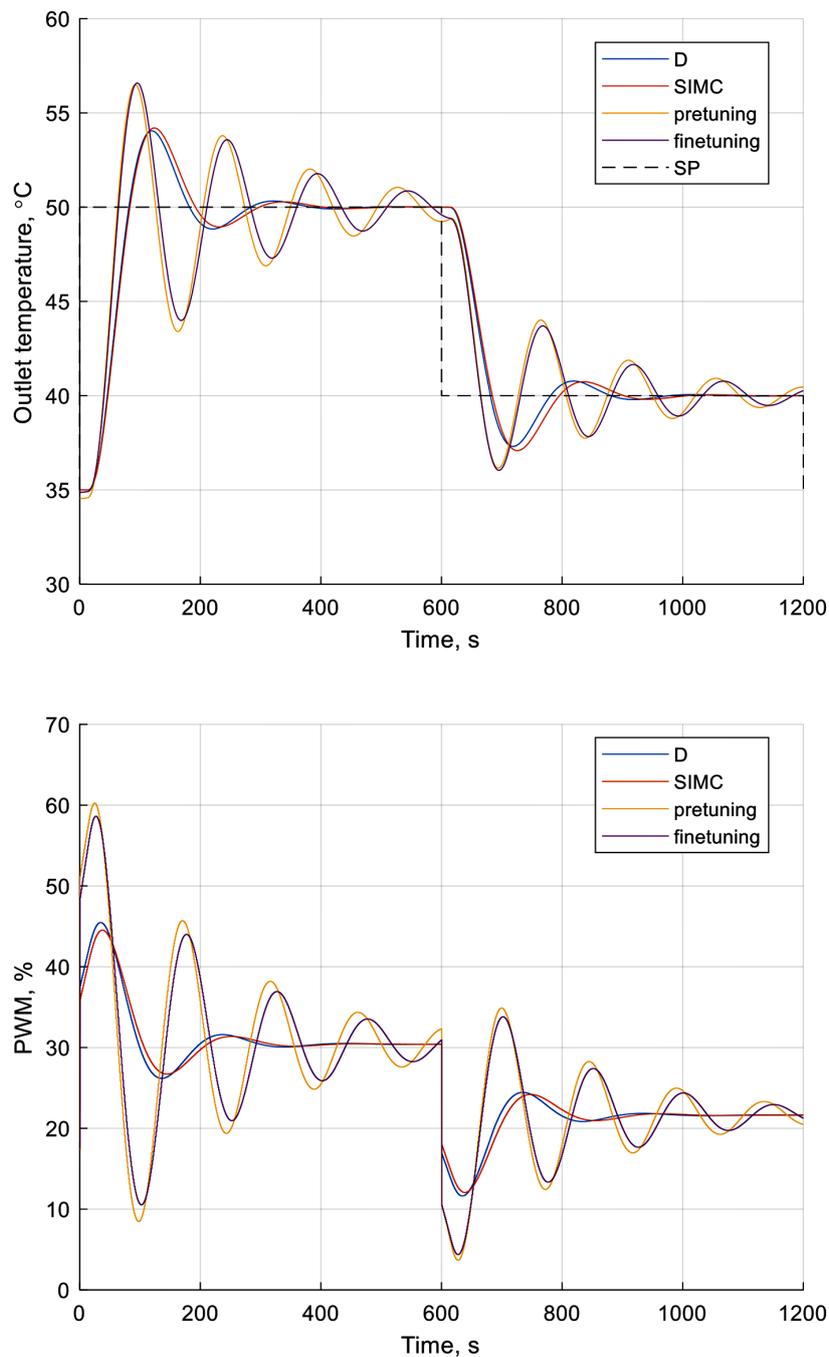


Figure 9. System response and control signal for change of setpoint. Flow set at 1.5 l/min

Table 3. Control performance indices for Selected tuning methods. results for changing the setpoint from 35 °C to 50 °C and for a flow rate of 2 L/min (Figure 8)

| Tunings | Indices | | | |
|------------|---------------|--------------|--------------|---------------|
| | Max overshoot | IAE | ITAE | Settling time |
| D | 1.28 | 869.5 | 39744 | 83.1 |
| SIMC | 1.85 | 877.1 | 36376 | 84.6 |
| Pretuning | 5.17 | 1010.9 | 80803 | 281.7 |
| Finetuning | 5.61 | 992.5 | 74745 | 183.6 |

Table 4. Control performance indices for selected tuning methods. Results for changing the setpoint from 50 °C to 40 °C and for a flow rate of 2 l/min (Figure 8)

| Tunings | Indices | | | |
|------------|---------------|--------------|--------------|---------------|
| | Max overshoot | IAE | ITAE | Settling time |
| D | 1.29 | 588.5 | 26762 | 81.8 |
| SIMC | 1.57 | 587.0 | 24504 | 83.3 |
| Pretuning | 4.35 | 677.3 | 54025 | 178.4 |
| Finetuning | 4.75 | 663.4 | 49791 | 181.1 |

Table 5. Control performance indices for selected tuning methods. Results for changing the setpoint from 35 °C to 50 °C and for a flow rate of 1.5 l/min (Figure 9)

| Tunings | Indices | | | |
|------------|---------------|---------------|---------------|---------------|
| | Max overshoot | IAE | ITAE | Settling time |
| D | 8.09 | 1152.5 | 8093.8 | 237.5 |
| SIMC | 8.4 | 1188 | 8584.1 | 241.2 |
| Pretuning | 12.99 | 1813.9 | 28280 | 534.1 |
| Finetuning | 13.17 | 1753.2 | 263534 | 485.4 |

Table 6. Control performance indices for selected tuning methods. Results for changing the setpoint from 50 °C to 40 °C and for a flow rate of 1.5 l/min (Figure 9)

| Tunings | Indices | | | |
|------------|---------------|--------------|--------------|---------------|
| | Max overshoot | IAE | ITAE | Settling time |
| D | 6.73 | 764.3 | 52937 | 164.6 |
| SIMC | 7.28 | 817.5 | 61054 | 178.1 |
| Pretuning | 9.60 | 1113.6 | 17112 | 467.2 |
| Finetuning | 9.9 | 1078.5 | 16010 | 411 |

methods are minor, although in most cases, the D method ensures better closed loop performance quantified by the considered indicators.

In order to improve the performance of the control system different flow rate F , the best solution would be to determine the FOPDT process approximation for the new operating point and repeat the cloud-based auto-tuning procedure or to used more complex nonlinear model of the real process that incorporates how the dynamics varies

according to the variations of the operating point. In both cases it is also possible to obtain gain-scheduling functionality that can be used for final improvement of the closed-loop performance.

CONCLUSIONS

In general, this paper presented the concept of the system that supports the auto-tuning

functionality available in PID function blocks implemented in Programmable Logic Controllers. The procedure is implemented using cloud-based resources and the concept was validated using Siemens equipment and programming environment

Validation of the proposed concept has been completed using virtual commissioning approach with the dynamical system represented by the electric flow heater with the higher-order dynamics and nonlinearities represented by time-invariance of the model. The proposed autotuning method was based on the results obtained during the experiment carried out for PID autotuning procedure built-in the PID function block. On the basis of the collected process response, the FOPDT approximation is derived in cloud environment and then a set of tunings method are used for PID tuning. The selected tuning methods include classical methods and the methods that are based on minimization of the selected closed loop performance indices. In each case, the performance of each closed-loop system with PID controller tuned by the considered method was quantified by the control performance indices, such as maximal overshoot, integral absolute error, integral time absolute error and settling time. The optimization-based method proposed in this work ensures significant improvement not only over the built-in autotuning methods but also over the classical tuning methods based directly on FOPDT process approximation. To summarize, the results presented in this paper show that a significant improvement in the closed-loop performance can be obtained even using very simple FOPDT approximation of the process dynamics, comparing to the built-in autotuning functionalities.

The concept was validated using Siemens equipment and programming environment but it can be extended to other industrial platforms. Its functionality can be limited to (more or less) advanced PID tuning methods that can be used for its auto-tuning based on the simple FOPDT approximation of the real process dynamics. This approach can be suitable for time-invariant processes. The proposed auto-tuning system can be also extended to using more complex process models (including very detailed physical models) and providing more complex functionalities (e.g. determining gain-scheduling technique).

Acknowledgments

This work was financed by The National Centre for Research and Development, Poland (NCBR) under the project “Lider XII” program (LIDER/38/0199/L-12/20/NCBR/2021) (M. Fraczak, P. Nowak, P. Grelewicz).

REFERENCES

1. Malinauskaite J. et al., Energy efficiency in the industrial sector in the EU, Slovenia, and Spain, *Energy*, 2020; 208: 118398, <https://doi.org/10.1016/j.energy.2020.118398>
2. Åström K. J. and Hägglund T., *PID Controllers: Theory, Design, and Tuning*. ISA, 1995.
3. Hang C. C. and Sin K. K., On-line auto tuning of PID controllers based on the cross-correlation technique, *IEEE Trans. Ind. Electron.*, Dec. 1991; 38(6): 428–437, <https://doi.org/10.1109/41.107098>
4. Killingsworth N. and Krstić M., Auto-tuning of PID controllers via extremum seeking, in *Proc. Amer. Control Conf.*, Portland, OR, USA, 2005; 2251–2256, <https://doi.org/10.1109/ACC.2005.1470304>
5. Liu F. and Er M. J., Design for auto-tuning PID controller based on genetic algorithms, in *Proc. 4th IEEE Conf. Ind. Electron. Appl.*, Xi'an, China, 2009; 1924–1928, <https://doi.org/10.1109/ICIEA.2009.5138538>
6. Ray P. P., An introduction to dew computing: Definition, concept and implications, *IEEE Access*, 2018; 6: 723–737, <https://doi.org/10.1109/ACCESS.2017.2775042>
7. Grelewicz P., Khuat T. T., Cieczot J., Nowak P., Kłopot T. and Gabrys B. Application of machine learning to performance assessment for a class of PID-based control systems, *IEEE Trans. Syst., Man, Cybern.: Syst.*, Jul. 2023; 53(7): 4226–4238, <https://doi.org/10.1109/TSMC.2023.3244714>
8. Thangatamilan M. et al., Industrial IoT (IIoT) based control of robotic arm integrated with PLC and HMI, in *Proc. ICSCDS*, Erode, India, 2023; 1208–1213, <https://doi.org/10.1109/ICSCDS56580.2023.10104636>
9. Borase R. P. et al., A review of PID control, tuning methods and applications, *Int. J. Dyn. Control*, 2021; 9: 818–827, <https://doi.org/10.1007/s40435-020-00665-4>
10. Sanchis R., Miguel-Escrig O., and Romero-Pérez J. A., Experimental tuning of PI controllers with symmetric send-on-delta sampling from the step response, *Int. J. Control*, 2023; 97(1): 101–110, <https://doi.org/10.1080/00207179.2023.2280138>
11. Sanchis R. and Peñarrocha-Alós I., Optimal tuning

- of PID controllers with derivative filter for stable processes using three points from the step response, *ISA Trans.*, Dec. 2023; 143: 596–610, <https://doi.org/10.1016/j.isatra.2023.10.009>
12. Kim J., Lee F., Lee J., and Kwon J., Nonlinear second-order plus time-delay model identification and nonlinear PID controller tuning based on extended linearization method, *Control Eng. Pract.*, 2024; 152: 106044, <https://doi.org/10.1016/j.conengprac.2024.106044>
 13. García Vázquez C. A. et al., Optimal tuning of PID controllers focused on energy efficiency in a multi-variable and coupled system: AHU case study, *Appl. Therm. Eng.*, 2025; 268, pt. B: 126053, <https://doi.org/10.1016/j.applthermaleng.2025.126053>
 14. Nguyen V.-K. et al., Design and implementation of fuzzy-based fine-tuning PID controller for PLC, *Int. J. Integr. Eng.*, 2024; 16(5): 359–372.
 15. Bujgoi G. and Sendrescu D., Tuning of PID controllers using reinforcement learning for nonlinear system control,” *Processes*, 2025; 13: 735, <https://doi.org/10.3390/pr13030735>
 16. Altinkaya H. and Ekmekci D., Tuning of PID controller in PLC-based automatic voltage regulator system using adaptive artificial bee colony–fuzzy logic algorithm, *Electronics*, 2024; 13: 5039, <https://doi.org/10.3390/electronics13245039>
 17. Kim K. H., Bae J. E., Chu S. C., and Sung S. W., Improved continuous-cycling method for PID autotuning, *Processes*, 2021; 9: 509, <https://doi.org/10.3390/pr9030509>
 18. Gao X.-T., Shen Y.-Y., and Huang C.-Q., Auto-tuning of filtered PID controller for industrial processes under routine operating conditions, *ISA Trans.*, 2025; 157: 186–198, <https://doi.org/10.1016/j.isatra.2024.11.051>
 19. Rehan A., Boiko I., and Zweiri Y., Autotuning of PID controller using the modified relay feedback test and optimization under uncertainty for robotic manipulators, *IET Control Theory Appl.*, 2024; 18(5): 581–602, <https://doi.org/10.1049/cth2.12592>
 20. Zeng D. et al., Research on improved auto-tuning of a PID controller based on phase angle margin, *Energies*, 2019; 12: 1704, <https://doi.org/10.3390/en12091704>
 21. Awtoniuk M., Sałat R., Worwa M., and Reshetiuk V., Implementation of PID autotuning procedure based on doublet-pulse method in PLC controller, *Adv. Sci. Technol. Res. J.*, 2024; 18(1): 89–97, <https://doi.org/10.12913/22998624/176832>
 22. Sanchis R. and Peñarrocha I., A PID tuning approach to find the optimal compromise among robustness, performance and control effort: Implementation in a free software tool, *Int. J. Control*, 2021; 97(1): 16–35, <https://doi.org/10.1080/00207179.2021.1989491>
 23. Slavíček L., Balda P., and Schlegel M., RexTuner: A new software tool for PID controller tuning in Siemens PLC systems, in *Proc. PC 2025*; 1–6, <https://doi.org/10.1109/PC65047.2025.11047421>
 24. Somefun O., Akingbade K., and Dahunsi F., The dilemma of PID tuning, *Annu. Rev. Control*, 2021; 52, <https://doi.org/10.1016/j.arcontrol.2021.05.002>
 25. Çelik D. et al., Advancements in nonlinear PID controllers: A comprehensive review, *Comput. Electr. Eng.*, 2026; 129, pt. A: 110775, <https://doi.org/10.1016/j.compeleceng.2025.110775>
 26. Fratzczak M., Nowak P., Kłopot T., Czeczot J., Bysko S. and Opilski B., Virtual commissioning for the control of continuous industrial processes: Case study, in *Proc. MMAR, Międzyzdroje, Poland*, 2015; 1032–1037, <https://doi.org/10.1109/MMAR.2015.7284021>
 27. O’Dwyer A., *Handbook of PI and PID Controller Tuning Rules*. Imperial College Press, 2009.
 28. Lagarias J. C., Reeds J. A., Wright M. H., and Wright P. E., Convergence properties of the Nelder–Mead simplex method in low dimensions, *SIAM J. Optim.*, 1998; 9: 112–147.
 29. Chien K. L., Hrones J. A., and Reswick J. B., On the automatic control of generalized passive systems, *J. Fluids Eng.*, Feb. 1952; 74(2): 175–183, <https://doi.org/10.1115/1.4015724>
 30. Skogestad S. and Grimholt C., The SIMC method for smooth PID controller tuning, in *PID Control in the Third Millennium*, 2012, https://doi.org/10.1007/978-1-4471-2425-2_5
 31. Hägglund T. and Åström K. J., Revisiting the Ziegler–Nichols tuning rules for PI control, *Asian J. Control*, 2002; 4(4): 364–380.
 32. Visioli A. and Sánchez-Moreno J., IMC-based tuning of PIDA controllers: A comparison with PID control, *IFAC-PapersOnLine*, 2024; 58(7): 1–6, <https://doi.org/10.1016/j.ifacol.2024.08.001>
 33. Veronesi M. and Visioli A., Optimized tuning rules for proportional–integral controllers with selected robustness, *Asian J. Control*, 2023; 25(3): 1731–1744, <https://doi.org/10.1002/asjc.2888>
 34. Shi G. et al., A controller synthesis method to achieve independent reference tracking performance and disturbance rejection performance, *ACS Omega*, 2022; 7: 16164–16186, <https://doi.org/10.1021/acsomega.2c01524>