Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

A novel fuzzy logic scheme for PID controller auto-tuning

Ilija Kamenko, Velimir Čongradac & Filip Kulić

To cite this article: Ilija Kamenko, Velimir Čongradac & Filip Kulić (2022) A novel fuzzy logic scheme for PID controller auto-tuning, Automatika, 63:2, 365-377, DOI: 10.1080/00051144.2022.2043988

To link to this article: https://doi.org/10.1080/00051144.2022.2043988

Published online: 23 Feb 2022.

AUTOMATIKA
2022, VOL. 63, NO. 2, 365–377
https://doi.org/10.1080/00051144.2022.2043988

OPEN ACCESS

A novel fuzzy logic scheme for PID controller auto-tuning

Ilija Kamenko, Velimir Čongradac and Filip Kulić

Faculty of Technical Sciences, Department of Computing and Control Engineering, University of Novi Sad, Novi Sad, Serbia

ABSTRACT

This paper presents a novel method for PID (proportional–integral–derivative) controller auto-tuning based on expert knowledge incorporated into a fuzzy logic inference system. The proposed scheme iteratively tries to improve the performance of the closed-loop system. As performance measures, the proposed scheme uses the characteristics of the step response (rise time, overshoot, and settling time). PID parameters in the first iteration can be calculated based on the basic open-loop step response experiment or it is possible to use current parameters. In each successive iteration, step response characteristics are measured and the relative changes expressed in the percentage of value in the first iteration are calculated and converted into linguistic values. The fuzzy expert system computes fuzzy values that are used after defuzzification as multiplying factors for current PID parameters. To achieve a balance between the aggressive and robust closed-loop response, as well as between the slower and the faster one, the fuzzy expert system works in three operating modes: the one for speeding up the system, the one for reducing the overshoot, and the one for a balanced reduction of rise time and overshoot. The performance and robustness are verified by computer simulation using an extensive range of different processes.

ARTICLE HISTORY
Received 3 August 2020
Accepted 10 September 2021

KEYWORDS
PID; fuzzy logic; expert knowledge; auto-tuning

1. Introduction

All automatic control systems are characterized by a specific control law or control laws. For basic control laws, their effect may linearly depend on the error (P law), integral of error (I law), or the first derivative of the error in time (D law). Combining basic control laws renders the PID (proportional–integral–derivative) law that is the most commonly used control law today due to its simple implementation in modern programmable logic controllers used in industry, as well as home and building automation. The PID controller has three adjustable parameters, one for each control law. The values of the parameters are closely related to closed-loop system performance and robustness. However, in addition to its simple implementation, setting up three adjustable PID parameters is not an easy task and has posed a challenge for many authors for years.

The first systematic way to tune PID parameters was proposed by Ziegler and Nichols in 1942 [1]. Inspired by this work, many authors suggested improvements to meet different performance or robustness requirements [2,3]. These methods are given in the form of simple expressions that formalize previously experimentally determined characteristics of process dynamics. Despite its imperfection, the original Ziegler and Nichols method is still widely used today because of its simplicity and ease of use. In the early 2000s, more than 60 years later than the original work, Åström and Hägglund proposed a revitalization of the original method in [4,5] while maintaining simplicity but improving performance for a broader set of different processes. Not long after, Šekara and Mataušek [6] also suggest a revitalization of the original Ziegler and Nichols work. A comprehensive overview of developments in PID controllers theory and practice can be found in [7], while a summary of tuning methods in a systematic way is given in [8].

To reduce the time and knowledge necessary for tuning PID parameters, Åström and Hägglund proposed the first automatic tuning method [9]. Such automatic tuning was accomplished by combining the relay experiment as a method for determining process dynamics with methods for computing the parameters of a PID controller. In recent years, the authors proposed different approaches to the automatic tuning of PID controller parameters [10–15]. The most recent comparison of PID auto-tuners is presented in [16,17].

After a parameter of the PID controller is tuned, it is often necessary to manually fine-tune parameters. The tuning is achieved by applying little corrections to the parameters based on practical experience. Fuzzy logic has been recognized as an appropriate method for implementing the operator's experience into the PID controller design. Zadeh proposed a fuzzy set theory in 1965 as an extension of the classical notion of a set [18]. Inspired by this work, in 1974 Mamdani

CONTACT Ilija Kamenko kamenko@uns.ac.rs University of Novi Sad, Faculty of Technical Sciences, Department of Computing and Control Engineering, Trg Dositeja Obradovića 6, Novi Sad, Serbia

proposed the first application of the fuzzy set theory in the fuzzy logic control algorithm to synthesize the linguistic control mechanism of an experienced operator [19]. In the following years, fuzzy logic has been successfully used in many applications and has resulted in significant achievements [20–27]. A concise overview of fuzzy control research breakthrough results is given in [28], showing the evolution of this field. Tzafestas and Papanikopoulos were among the first ones to propose an approach in which the performance of the closed-loop system is improved by slightly changing the values of PID parameters based on a fuzzy matrix that contains the experience of an operator in the condensed rule form [29]. The authors in [30] proposed an auto-tuning algorithm based on a fuzzy inference mechanism for tuning PID parameters online using the response error signal of the closed-loop system as input. Many authors have applied a similar principle and introduced slightly different schemes to implement expert knowledge in the PID tuning process by using the fuzzy logic concept [31–39]. A comparison of fuzzy logic auto-tuning methods is presented in [40].

The purpose of this study is to develop a novel scheme for automatic tuning of a PID controller based on expert knowledge incorporated into a fuzzy inference system. The primary aim of the scheme is to enhance the performance of the arbitrary closed-loop system while maintaining robustness within acceptable limits with minimal investment of time, effort, and required knowledge.

The novelty of the approach is that the scheme in an iterative procedure modifies the parameters of the PID controller based on response characteristics. Unlike other previously mentioned methods [30,31,36] that use the error signal and the first derivate of the error signal, this method uses the rise time, overshoot, and the settling time as input in the fuzzy expert system. In addition, this method works offline as the classical auto-tuning method and is not intended for continuous adaptation, unlike most methods that utilize the fuzzy concept. Besides, this approach makes it possible to select an operating mode and, in that way, make an influence on setting performance and robustness tuning goals.

The motivation for this research comes from the real problems that the authors encountered during the implementation of the automatic control system over the years in industry and building automation.

2. Classical tuning methods

The well-known expression of a PID controller is the following:

$$u(t) = K_p\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de(t)}{dt}\right) \qquad (1)$$

where *e(t)* = *r(t)-y(t)* is a system error signal that represents the difference between setpoint value *r(t)* and current system output value *y(t)*; *u(t)* is the control variable, K_p is a proportional gain, T_i is an integral time constant, and T_d is a derivative time constant.

Classical tuning methods like Ziegler and Nichols and related methods are model-based methods. These methods are based on the approximation of process dynamics using a simple model. A simple process model can be obtained from an open-loop response. The first-order process with time delay (FOPTD) process model that is commonly used in process control has the following transfer function:

$$G(s) = \frac{K}{1+sT}e^{-sL} \qquad (2)$$

where K is the static gain, T is time constant (lag), and L is time delay. There are some processes with the dynamics that contain integration or very long-time constants and that will not reach a steady state under open-loop conditions. A combination of an integrator and a time delay makes the first-order integral process with a time delay (FOIPTD) process model with the transfer function:

$$G(s) = \frac{K_v}{s}e^{-sL} \qquad (3)$$

where K_v is velocity gain and L is the time delay. The parameters of the PID controller are then expressed in terms of the obtained process parameters using simple formulas [8].

The present research adopted the revitalization method proposed by Åström and Hägglund [5] as an initial tuning for a new proposed fuzzy expert system method for PID parameter auto-tuning. This method was first used in [4] for PI controller and the same approach was later used for PID control with additional constraints in [5]. For simplicity, the Åström and Hägglund method will be denoted as the A-H method in the rest of this paper. For stable processes, according to this method, PID parameters can be expressed as:

$$K_p = \frac{1}{K}\left(0.2 + 0.45\frac{T}{L}\right) \qquad (4)$$

$$T_i = \frac{0.4L + 0.8T}{L + 0.1T}L \qquad (5)$$

$$T_d = \frac{0.5LT}{0.3L + T} \qquad (6)$$

For integrating processes, Equations (4)-(6) can be written as:

$$K_p = \frac{0.45}{K_v} \qquad (7)$$

$$T_i = 8\mathrm{L} \qquad (8)$$

$$\mathrm{T}_d = 0.5L \qquad (9)$$

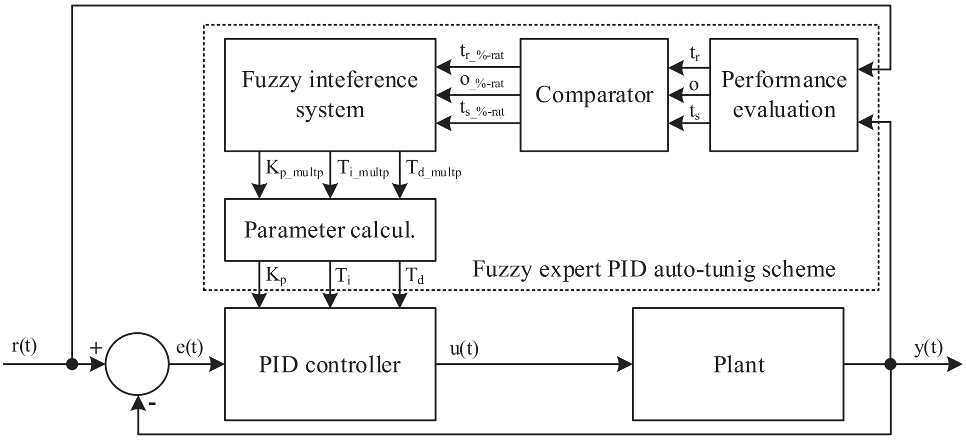

Figure 1. Block diagram of the proposed FES scheme.

3. Proposed fuzzy expert system

A block diagram of the proposed fuzzy expert system (FES) for auto-tuning PID controller parameters is shown in Figure 1.

It consists of the performance evaluation part, comparator, fuzzy interference system, and parameter calculation part. The main purpose of the performance evaluation part is to evaluate the closed-loop step response based on current value *y(t)* and setpoint value *r(t)*. The closed-loop response is evaluated using rising time t_r (time from 10% to 90% steady-state value), overshoot *o,* and settling time t_s (time of entering in 2% region of steady-state value) as performance measures. The comparator compares the current values of performance measures with reference values that are saved in the first iteration. The comparator renders the percentage ratios of rise time $t_{r\_\%-rat}$, overshoot $o_{\_\%-rat}$, and settling time $t_{s\_\%-rat}$, which can be expressed as follows:

$$t_{r\_\%-rat} = \frac{t_r}{t_{r\_ref}} 100\% \tag{10}$$

$$o_{\_\%-rat} = \frac{o}{o_{\_ref}} 100\% \tag{11}$$

$$t_{s\_\%-rat} = \frac{t_s}{t_{s\_ref}} 100\% \tag{12}$$

The fuzzy interference system based on percentage ratios calculates PID parameter multipliers K_{p_multp}, T_{i_multp}, and T_{d_multp} using expert knowledge within the rule base and fuzzy inference mechanism. The parameter calculation part calculates new PID parameters K_p, T_i, and $T_d$using multipliers calculated in the previous part and the current value of PID parameters, according to the following equations:

$$K_p = K_p + K_p K_{p\_multp} \tag{13}$$

$$T_i = T_i + T_i T_{i\_multp} \tag{14}$$

$$T_d = T_d + T_d T_{d\_multp} \tag{15}$$

3.1. Test batch

The test batch consists of 133 processes whose transfer functions are given in Equations (16)–(24). The test batch is discussed in [5] and it is here used practically unchanged. The test batch is used to derive expert knowledge and test final solutions. The test batch includes delay dominated, lag dominated, and integrating processes.

$$\begin{aligned} G_1(s) &= \frac{e^{-s}}{1+sT}, \\ T = {} & 0.02,\ 0.05,\ 0.1,\ 0.2, 0.3,\ 0.5, \\ & 0.7,\ 1,\ 1.3,\ 1.5,\ 2,\ 4,\ 6,\ 8, 10, \\ & 20,\ 50,\ 100,\ 200,\ 500,\ 1000 \end{aligned} \tag{16}$$

$$\begin{aligned} G_2(s) &= \frac{e^{-s}}{(1+sT)^2}, \\ T = {} & 0.01,\ 0.02,\ 0.05,\ 0.1,\ 0.2, \\ & 0.3,\ 0.5,\ 0.7,\ 1, 1.3,\ 1.5, \\ & 2,\ 6,\ 8,\ 10,\ 20,\ 50,\ 100, \\ & 200,\ 500 \end{aligned} \tag{17}$$

$$\begin{aligned} G_3(s) &= \frac{1}{(1+s)(1+sT)^2}, \\ T = {} & 0.005,\ 0.01,\ 0.02, \\ & 0.05,\ 0.1,\ 0.2,\ 0.5, \\ & 2,\ 5,\ 10 \end{aligned} \tag{18}$$

$$G_4(s) = \frac{1}{(1+s)^n}, \quad n = 3,\ 4,\ 5,\ 6,\ 7,\ 8 \tag{19}$$

$$\begin{aligned} G_5(s) &= \frac{1}{(1+s)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)}, \\ & \alpha = 0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9 \end{aligned} \tag{20}$$

$$\begin{aligned} G_6(s) &= \frac{e^{-sL}}{s(1+Ts)}, \\ T &= 0.01,\ 0.02,\ 0.05,\ 0.1,\ 0.3,\ 0.5,\ 0.7, \end{aligned}$$

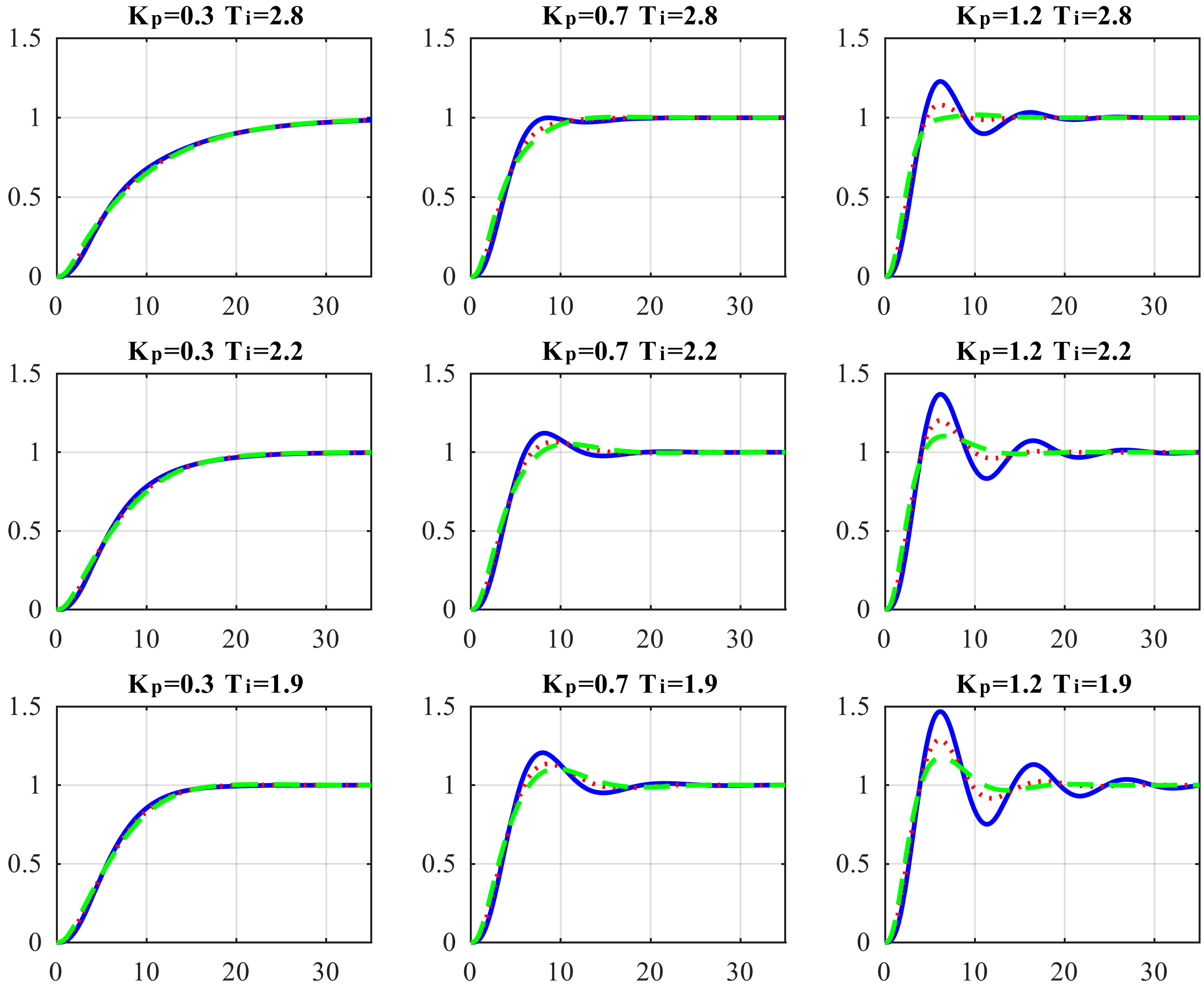

Figure 2. Tuning map for PID control of a process G_4 (19) with parameter $n = 4$ (blue solid $T_d = 0.2$; red dotted $T_d = 0.7$; green dashed $T_d = 1.1$).

$$0.9,\ 1.0\ T + L = 1 \tag{21}$$

$$G_7(s) = \frac{T}{(1+sT)(1+sT_1)} e^{-sL},$$

$$L = 0.01,\ 0.02,\ 0.05,\ 0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.9,\ 1.0$$

$$T = 1,\ 2,\ 5,\ 10 \qquad T_1 + L = 1 \tag{22}$$

$$G_8(s) = \frac{1+\alpha s}{(1+s)^3},$$

$$\alpha = 0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.6,\ 0.7,\ 0.8,$$

$$0.9,\ 1.0,\ 1.1 \tag{23}$$

$$G_9(s) = \frac{1}{(1+s)((sT)^2 + 1.4sT + 1))},$$

$$T = 0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9,\ 1.0 \tag{24}$$

3.2. Expert knowledge

Since many methods give only rouge values, it is often necessary to apply manual tuning to obtain better closed-loop properties. This is typically done through experiments with the closed-loop process. The closed-loop response is observed and controller parameters are modified. The modifications are based on simple rules that can be expressed as tuning maps. The tuning maps are a matrix of transient responses corresponding to systematic variations in PID parameters whose essential function is to provide insight into how changes in controller parameters influence the behaviour of the closed-loop system.

To speed up the synthesis of expert knowledge without losing the generality, only a few processes from the test batch given in Equations (16)-(24) are observed. Each process represents one of the four ways to classify the processes of the test batch. The systems with the integrator are classified into the first group represented by process G_6 given in Equation (21) with parameter $L = 0.1$. Process G_5 given in Equation (20) with parameter $\alpha = 0.5$ represents the second group of lag-dominated processes. The third group includes the balanced lag and delay processes represented by process G_4 given in Equation (19) with parameter $n = 4$. Process G_2 given in Equation (17) with parameter $T = 0.02$ represents the last group of delay-dominated processes.

Tuning maps are created for all four processes. As an example, a tuning map for process G_4 given by Equation (18) with parameter $n = 4$ is shown in Figure 2. The

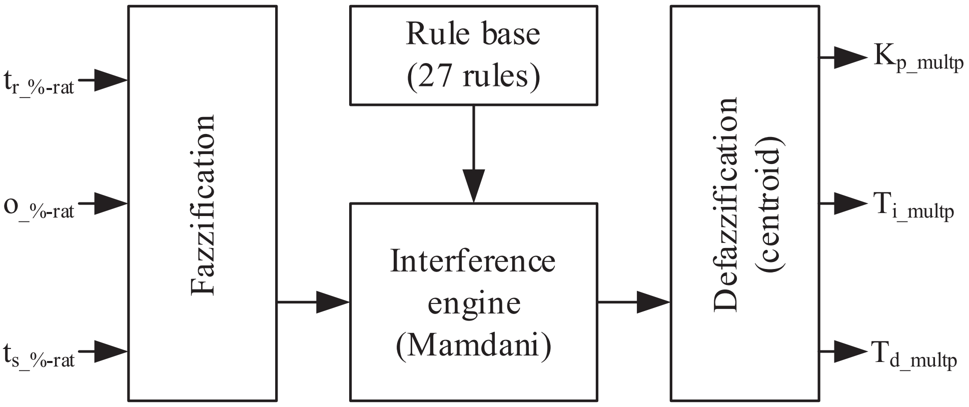

Figure 3. Block diagram of a fuzzy interference system.

Table 1. Basic relations between increasing PID controller parameters and step response characteristics.

Parameter	Rise time	Overshoot	Settling time
K_p	decrease	increase	minor decrease
T_i	increase	decrease	increase/decrease
T_d	minor increase	decrease	decrease

figure illustrates how step response is influenced by changes in the proportional gain (K_p), integral time (T_i), and derivative time (T_d). Parameter K_p increases along abscissa and parameter T_i along the ordinate, while the response for each value of parameter T_d is represented with different line types and colours.

Based on the created tuning maps it is possible to derive the relations between the increasing PID controller parameter and the values of step response characteristics (rise time, overshoot, and settling time). The derived fundamental relations are given in Table 1.

3.3. *Fuzzy interference system*

This study adopted the Mamdani fuzzy interference system with three inputs and three outputs. A block diagram of the proposed system is shown in Figure 3.

The rise time ratio $t_{r\_\%-rat}$, overshoot ratio $o_{\_\%-rat}$, and settling time ratio $t_{s\_\%-rat}$ were chosen as input parameters. The value of input parameters was set to range from 0% to 200%. For the ratios greater than 200%, the values of current measures have already been certainly larger than those of referent values, so the upper limit was saturated at level 200. The input variables have been decomposed into three different fuzzy linguistic levels: low (L), medium (M), and high (H). The shape of membership functions for all three inputs is shown in Figure 4.

Multipliers K_{p_multp}, T_{i_multp}, and T_{d_multp}were chosen as output parameters. The value of the output parameters was set to be in the range from −0.5 to 0.5. Therefore, PID parameters could be adjusted by up to 50% of their value up or down in each iteration. The output variables have been decomposed into seven different fuzzy linguistic levels: negative high (NH), negative medium (NM), negative low (NL), zero (Z), positive low (PL), positive medium (PM), and positive high (PH). The shape of membership functions for all three multipliers can be seen in Figure 5.

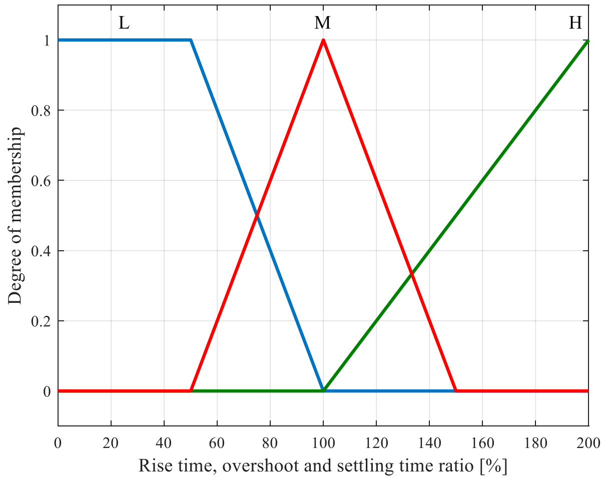

Figure 4. Membership functions for three inputs: rise time, overshoot, and settling time ratios.

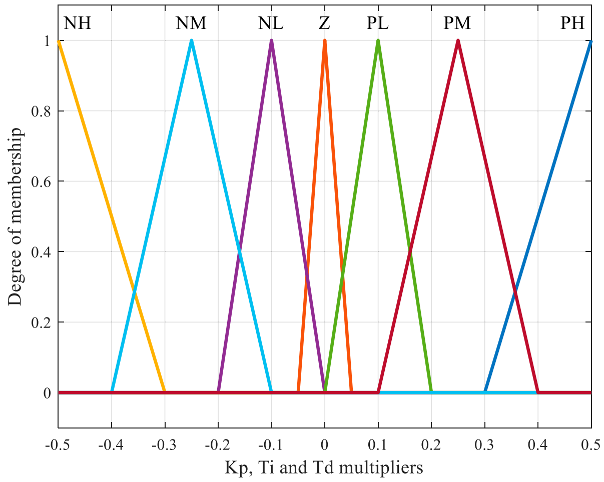

Figure 5. Membership functions for three outputs: K_p, T_i, and T_d multipliers.

A comparative analysis of relations obtained from tuning maps of four representative processes was used to create a rule base. Since each input is described

Table 2. The rule base of a fuzzy interference system.

Rule	$t_{r\_\%-rat}$	$o_{\_\%-rat}$	$t_{s\_\%-rat}$	K_{p_multp}	T_{i_multp}	T_{d_multp}
1	L	L	L	Z	Z	Z
2	L	L	M	Z	Z	Z
3	L	L	H	NL	NM	Z
4	L	M	L	NL	PM	PL
5	L	M	M	NL	PM	PL
6	L	M	H	NL	PM	NL
7	L	H	L	NL	PM	PL
8	L	H	M	NL	PM	PL
9	L	H	H	NL	NM	PL
10	M	L	L	PM	PL	NL
11	M	L	M	PM	PL	NL
12	M	L	H	PM	PL	NL
13	M	M	L	PM	PM	Z
14	M	M	M	PM	PM	Z
15	M	M	H	PM	PM	Z
16	M	H	L	NM	PH	PM
17	M	H	M	NM	PH	PM
18	M	H	H	NM	PH	PM
19	H	L	L	PH	NL	NL
20	H	L	M	PH	NL	NL
21	H	L	H	PH	NL	NL
22	H	M	L	PH	NM	NM
23	H	M	M	PH	NM	NM
24	H	M	H	PH	NM	NM
25	H	H	L	NH	NH	NH
26	H	H	M	NH	NH	NH
27	H	H	H	NH	PH	NH

with three fuzzy sets, a well-defined fuzzy inference system should have $3 \times 3\times3 = 27$ fuzzy rules. Each rule defines how PID parameters should be changed depending on the values of ratios between current and referent performance measures. The overview of fuzzy rules is shown in Table 2.

3.4. Operating modes

The trade-off between performance and robustness varies among different control problems, which is why it is desirable to have a design parameter to change the properties of the closed-loop system. The positions of membership functions of fuzzy sets L, M, and H for all three inputs have a considerable influence on FES operation, which can be used to select an objective for PID tuning. Triangular and trapezoidal membership functions can be expressed by a set of their vertexes. The position of the membership function depending on parameter P can be described by the following equations:

$$L = [0,\ 0,\ P,\ max((50 + P),\ 100)] \tag{25}$$

$$M = [P, max((50 + P), 100), max((100 + P), 150)] \tag{26}$$

$$H = [max((50 + P), 100),\ 200,\ 200] \tag{27}$$

Figure 6 shows the position of membership functions for two different values of parameter P. From Equations (25)-(27) it can be seen that when $P = 0$, membership function positions come down to the basic version shown in Figure 4. With the proper selection of parameter P, different operating modes of the proposed algorithm can be obtained. This study proposes three operating modes of the FES tuning method.

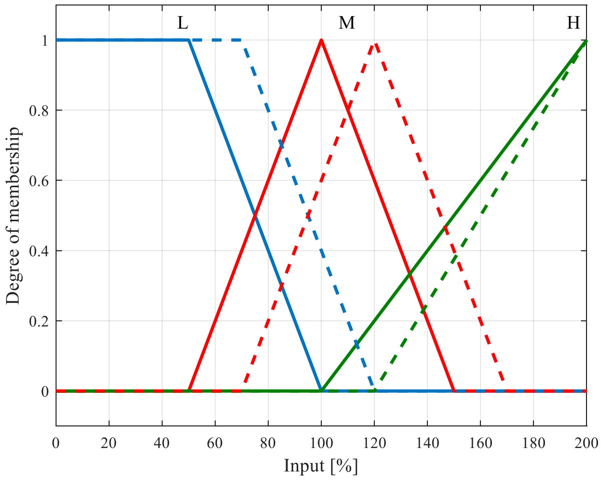

Figure 6. Positions of membership functions for two different values of a parameter P – solid lines $P = 50$, dashed lines $P = 75$.

The first operating mode is Speed up tuning, whose primary objective is to decrease the rise time of the closed-loop system in the first place, and then decrease the settling time and overshoot while keeping sensitivity within reasonable limits. Therefore, the value of parameter P for the first input $t_{r\_\%-rat}$ was set to 40, for the second input $o_{\_\%-rat}$ it was set to 100, and for the third input $t_{s\_\%-rat}$ it was set to 100.

The second operating mode is Reduce overshoot tuning, whose primary objective is to decrease the overshoot of the closed-loop system in the first place, and then decrease rise time and settling time while keeping sensitivity within reasonable limits. Therefore, the value of parameter P for the first input $t_{r\_\%-rat}$ was set to 100, for the second input $o_{\_\%-rat}$ it was set to 40, and for the third input $t_{s\_\%-rat}$ it was set to 100.

The third operating mode is Balanced tuning, whose primary objective is to balance the decrease overshoot and rise time of the closed-loop system in the first place, and then reduce the settling time while keeping sensitivity within reasonable limits. Therefore, the value of parameter P for each of three inputs ($t_{r\_\%-rat}$, $o_{\_\%-rat}$, and $t_{s\_\%-rat}$) was set to 75.

3.5. Tuning procedure

The flow chart of the proposed FES auto-tuning scheme is shown in Figure 7. If there are no initial PID parameters, or the closed-loop system has sluggish performance, new initial parameters can be calculated in the pre-tuning phase. Alternatively, pre-tuning can be skipped if closed-loop performances are satisfactory (Step 2).

In the pre-tuning phase, the PID controller is disconnected. Therefore, initial tuning is performed offline.

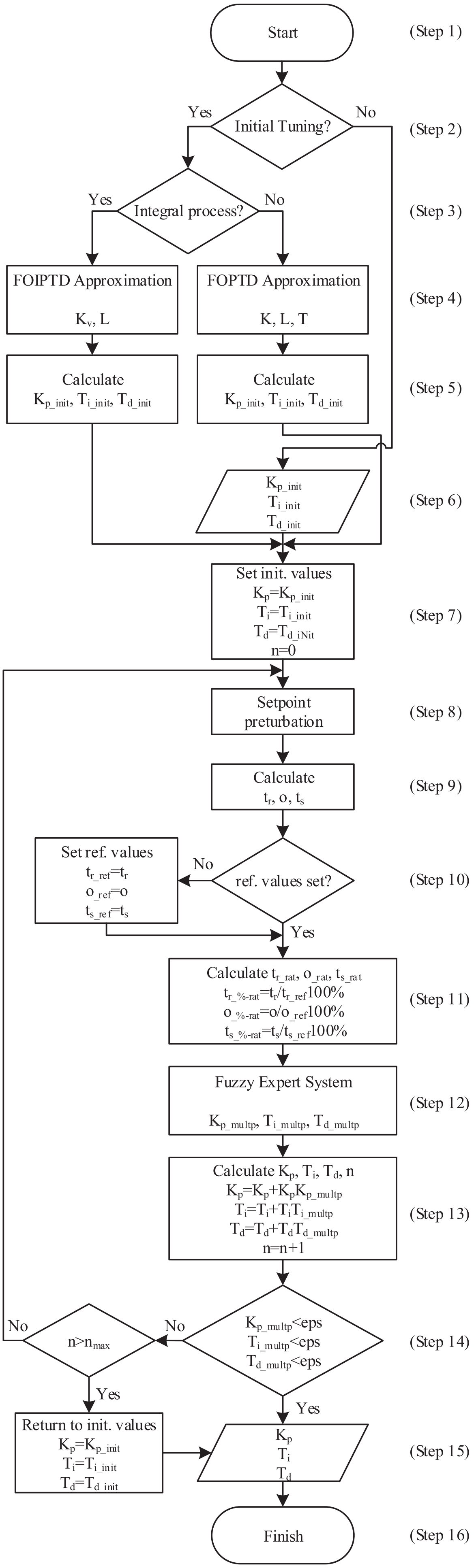

Figure 7. Flow chart of the proposed FES tuning method.

Based on the open-loop response, the system is approximated with the model given in Equations (2) or (3), depending on whether the system possesses an integrator (Steps 3-4). After successfully obtaining initial parameters (Step 6), the PID controller is connected again, and the second phase can begin.

The second phase is performed online, with initial parameters adopted as current parameters (Step 7).The first iteration of the fine-tuning phase includes setpoint perturbation (Step 8) and calculation of performance measures of the step response (Step 9). The computed performance measures are saved as referent values for this iteration and further ones (Step 10). The relative values of performance measures expressed as the percentage of saved referent values are calculated in Step 11. The fuzzy expert system generates multiplier factors for PID parameters using calculated relative values of performance measures as input in Step 12. Based on computed multiplier factors and current values of PID parameters, new parameters are calculated in Step 13. These steps are repeated until multipliers are greater than tolerance, and the number of iterations does not exceed the maximum number (Step 14). When multipliers become smaller than tolerance, the last PID parameters are adopted as final (Step 15) and the procedure is finished successfully (Step 16). If the number of iterations is exceeded, parameters are set back to initial values (Step 15) and the procedure ends unsuccessfully (Step 16). This stopping criterion based on the number of iterations is beneficial if the process does not converge.

4. Results and discussion

In addition to the performance measures used in the tuning process, additional performance and robustness measures are used to enable a better comparison of the obtained results. The integral of absolute error *IAE* is a beneficial measure given in:

$$IAE = \int_0^t |e(t)|dt \tag{28}$$

where *e(t)* = *r(t)-y(t)* is a system error that represents the difference between setpoint value *r(t)* and current system output value *y(t)*. The integral of the absolute error for load disturbance response IAE_d is the same as the previously defined measure but it is used in load disturbance response. Robustness is specified through the maximum value of the sensitivity function:

$$M_s = \max_{\omega} \left| \frac{1}{1 + G_c(j\omega)G(j\omega)} \right| \tag{29}$$

where G_c is the transfer function of the controller, while G is the transfer function of the process. Value M_s can be interpreted as the inverse of the shortest distance between the loop transfer function and the critical point in the Nyquist plot [5].

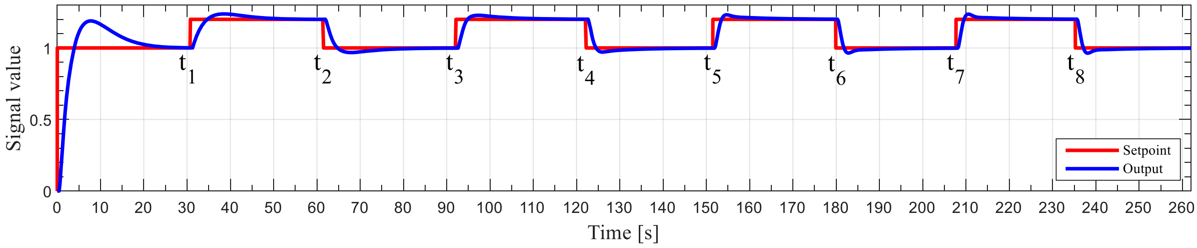

Figure 8. Procedure of tuning PID controller parameters for process G_6 (21) with parameter $T = 0.5$ using the proposed FES tuning method in the speed up tuning mode.

For all processes from the test batch given in Equations (16)-(24), excluding the four ones used for expert knowledge synthesis, PID parameters are initially tuned by the A-H method. The processes are then fine-tuned by the FES tuning method with all three modes.

The results obtained using both methods are compared below.

4.1. Fes tuning method in the speed up mode

Consider process G_6 (21) with parameter $T = 0.5$ as the first example. Fitting the model given in Equation (3) to the process, it was found that the apparent time delay and velocity gain were $T = 1$ and $K_v = 1$. By applying A-H tuning, the following PID parameters are obtained: $K_p = 0.45$, $T_i = 8$, and $T_d = 0.5$. The FES tuning method in the speed up mode is applied and the procedure through iterations can be seen in Figure 8. The procedure starts at moment t_1 and ends at moment t_8. After seven iterations, the following PID parameters are obtained: $K_p = 0.99$, $T_i = 10.83$, and $T_d = 0.36$.

Table 3. Overview of PID parameters through iterations of the proposed FES tuning method in the speed up tuning mode for process G_6 (21) with parameter T = 0.5.

Iteration	K_p	T_i	T_d
1	0.45	8.00	0.50
2	0.56	8.80	0.45
3	0.68	9.49	0.41
4	0.79	10.08	0.39
5	0.90	10.52	0.37
6	0.97	10.77	0.36
7	0.99	10.83	0.36

Table 3 shows how parameters are changed through iterations, from the first one to the last one. For all seven iterations, the responses of the system to step changes in setpoint and load are shown in Figure 9. From the figure, it can be seen that the rise time is shorter and shorter with each iteration. At the same time, the overshoot hardly changes while the rise time is slightly improved. It can also be seen that the response to the unit step changes in load improves in each subsequent iteration.

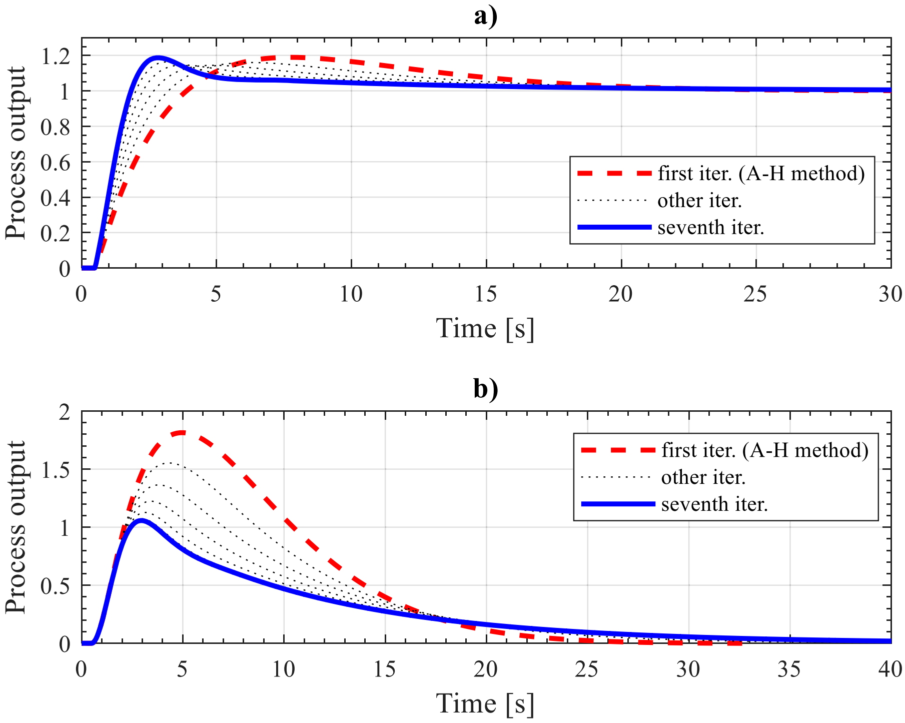

Figure 9. Response to a unit step change in setpoint (a) and load (b) for process G_6 (21) with parameter $T = 0.5$. The PID controller is tuned by the A-H method and then fine-tuned by the proposed FES tuning method in the speed up tuning mode.

Table 4. Performance and robustness of the PID controller for process G_6 (21) with parameter $T = 0.5$ tuned by the A-H method and fine-tuned by the FES tuning method in the speed up mode.

	t_r [s]	o [%]	t_s [s]	IAE	IAE_d	M_s
A-H method	2.52	18.89	20.77	3.67	17.79	1.22
FES tuning – speed up mode	1.01	18.69	17.58	2.24	10.93	1.62
Improvement [%]	−60.02	−1.10	−15.34	−38.98	−38.57	32.64

Table 5. Average improvement of performance and robustness measures for all processes achieved by the FES tuning method in the speed up mode after A-H initial tuning.

Process type	Count	t_r [%]	o[%]	t_s [%]	IAE [%]	IAE_d [%]	M_s [%]	Iterations
G_1 (16)	21	−60.92	−27.97	−18.99	−25.05	−24.24	20.48	3.57
G_2 (17)	20	−61.08	−17.80	−8.28	−32.94	−31.50	22.56	11.60
G_3 (18)	10	−59.64	−3.28	−11.65	−44.59	−31.46	20.98	23.50
G_4 (19)	5	−57.67	−2.19	72.44	−9.64	−8.93	46.05	49.00
G_5 (20)	8	−57.68	−21.02	37.60	−25.88	−16.66	25.85	37.25
G_6 (21)	8	−60.15	−14.21	−21.99	−42.47	−40.64	21.79	8.63
G_7 (22)	36	−60.35	−14.73	−39.89	−44.56	−42.00	18.94	10.75
G_8 (23)	11	−60.68	−48.21	21.78	−22.44	−18.58	36.72	19.09
G_9 (24)	10	−52.50	10.41	169.67	58.61	161.35	39.95	46.60
All	129	−59.64	−17.25	2.37	−27.06	−15.97	24.71	17.19

Performance and robustness measures are shown in Table 4 to enable a better comparison of results. It can be seen from the table that the proposed method meets the previously set goals but with the degradation of sensitivity, which is expected as the system speeds up.

The procedure is repeated for all other processes, and improvements are calculated. Average improvements grouped by process type are displayed in Table 5. The FES tuning method in the speed up mode achieved a significant average decrease of nearly 60% in rise time, which is the primary goal of this operating mode. At the same time, a visible average improvement of overshoot and settling time can also be observed, excluding a couple of groups of processes with the degradation of settling time. The values of the integral of the absolute error of the reference step response and load step response are also noticeably reduced on average. Speeding up the system degradation of sensitivity is expected and is almost 25% on average, which is satisfactory. Considering all the processes, the average duration of the procedure is about 17 iterations.

Table 6. Overview of PID parameters through iterations of the proposed FES tuning method in the reduce overshoot mode for process G_5 (20) with parameter $\alpha = 0.1$.

Iteration	K_p	T_i	T_d
1	4.29	0.44	0.05
2	3.86	0.55	0.06
3	4.20	0.64	0.06
4	4.76	0.71	0.06
5	5.38	0.74	0.05

4.2. *Fes tuning method in the reduce overshoot mode*

Consider process G_5 (20) with parameter $\alpha = 0.1$ as the second example. Fitting the model given in Equation (2) to the process, it was found that the apparent time delay and time constants were $L = 0.11$ and $T = 0.99$. Applying the A-H method, the following PID parameters are obtained: $K_p = 4.29$, $T_i = 0.44$, and $T_d = 0.05$. The FES tuning method in the reduce overshoot mode is applied and the procedure through iterations can be seen in Figure 10. The procedure starts at moment t_1 and ends at moment t_6. After five iterations, the following PID parameters are obtained: $K_p = 5.38$, $T_i = 0.74$, and $T_d = 0.05$. Table 6 shows how the parameters are changed through iterations, from the first one to the last one. For all five iterations, the responses of the system to step changes in the setpoint and load are shown in Figure 11. The figure shows that overshoot decreases throughout iterations. At the same time, the rise time changes little while the settling time shows visible improvement. It can also be seen that the response to unit step changes in load is improved.

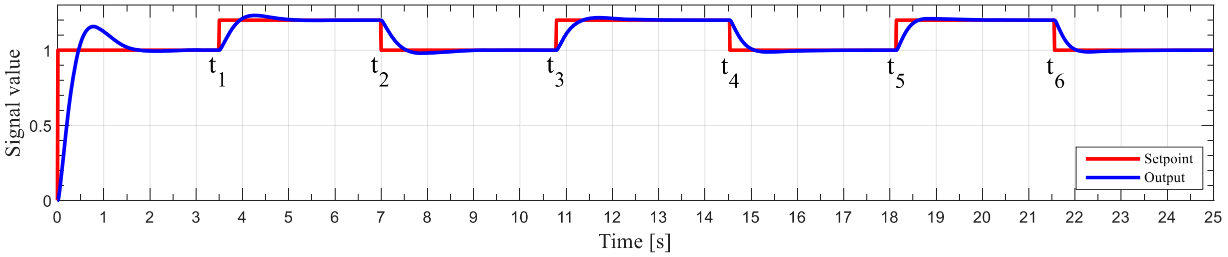

Figure 10. Procedure of tuning PID controller parameters for process G_5 (20) with parameter $\alpha = 0.1$ using the proposed FES tuning method in the reduce overshoot mode.

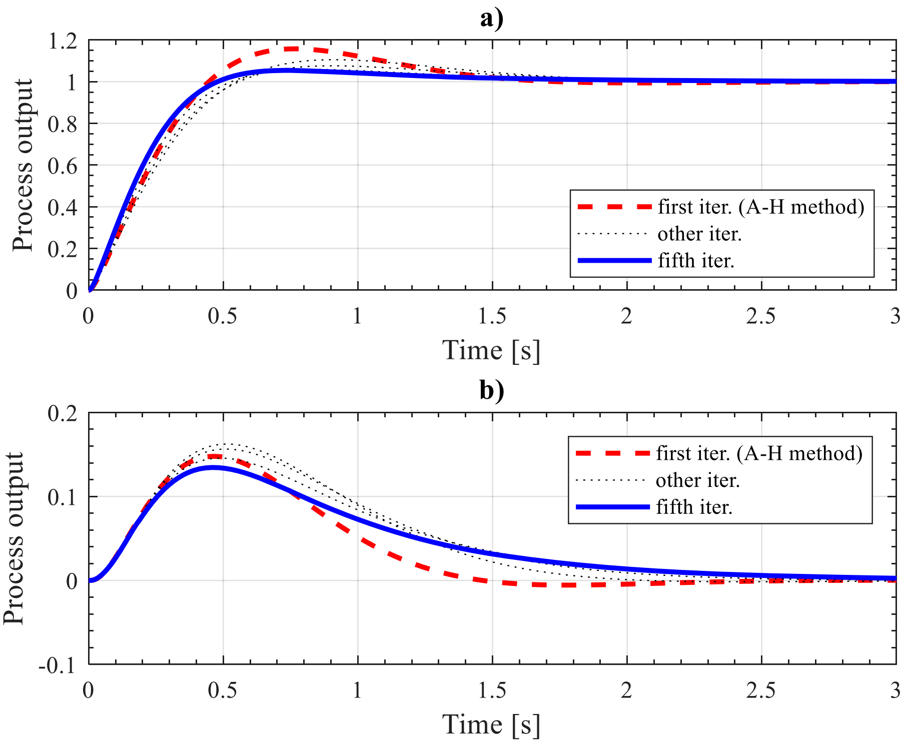

Figure 11. Response to a unit step change in setpoint (a) and load (b) for process G_5 (20) with parameter $\alpha = 0.1$. The PID controller is tuned by the A-H method and then fine-tuned by the FES tuning method in the reduce overshoot mode.

Table 7. Performance and robustness of the PID controller for process G_5 (20) with parameter $\alpha = 0.1$ tuned by the A-H method and fine-tuned by the FES tuning method in the reduce overshoot mode.

	t_r [s]	o [%]	t_s [s]	IAE	IAE_d	M_s
A-H method	0.32	15.74	1.50	0.31	0.11	1.14
FES tuning – red. overshoot mode	0.32	5.41	1.42	0.23	0.14	1.09
Improvement [%]	−1.35	−65.63	−5.31	−25.16	25.57	−3.90

Performance and robustness measures are shown in Table 7. It can be seen that the proposed method meets the previously set goals with a little improvement of sensitivity.

The procedure is repeated for all processes, and improvements are calculated. The results are grouped by process type and average improvements for all groups are displayed in Table 8. The FES tuning method in the reduce overshoot operating mode achieved a significant average decrease of nearly 64% in overshoot, which is the primary goal of this operating mode. At the same time, there is a minimal average improvement of rising time and settling time. The value of the integral of an absolute error on the reference step response is reduced by nearly 17% on average. At the same time, there is a small average degradation of the integral of an absolute error on the load step response and sensitivity. The average duration of the procedure considering all the processes is nearly ten iterations.

4.3. Fes tuning method in the balanced mode

Consider process G_7 (22) with parameter $T = 1$ and $L = 0.1$ as the third example. Fitting the model given in Equation (2) to the process, it was found that the apparent time delay and time constants were $L = 0.6$

Table 8. Average improvement of performance and robustness measures for all processes achieved by the FES tuning method in the reduce overshoot mode after A-H initial tuning.

Process type	Count	t_r [%]	o[%]	t_s [%]	IAE [%]	IAE_d [%]	M_s [%]	Iterations
G_1 (16)	21	−25.49	−25.96	14.60	−14.28	21.84	33.82	16.33
G_2 (17)	20	−19.54	−26.00	−4.50	−15.68	−7.02	26.37	13.50
G_3 (18)	10	−2.10	−72.66	−11.78	−21.52	21.53	−2.79	5.30
G_4 (19)	5	−3.33	−99.15	−18.30	−9.85	−5.52	1.38	4.00
G_5 (20)	8	−1.52	−90.85	−31.24	−17.49	−1.41	−2.35	4.13
G_6 (21)	8	−6.10	−57.55	53.03	−11.62	84.49	6.02	35.38
G_7 (22)	36	−4.97	−75.99	−22.13	−23.43	5.63	1.93	4.78
G_8 (23)	11	−3.84	−99.97	−15.33	−12.51	−7.24	5.35	4.27
G_9 (24)	10	−1.78	−94.33	−10.70	−14.39	1.03	−1.47	4.40
All	129	−9.80	−63.98	−6.91	−17.33	10.11	10.54	9.81

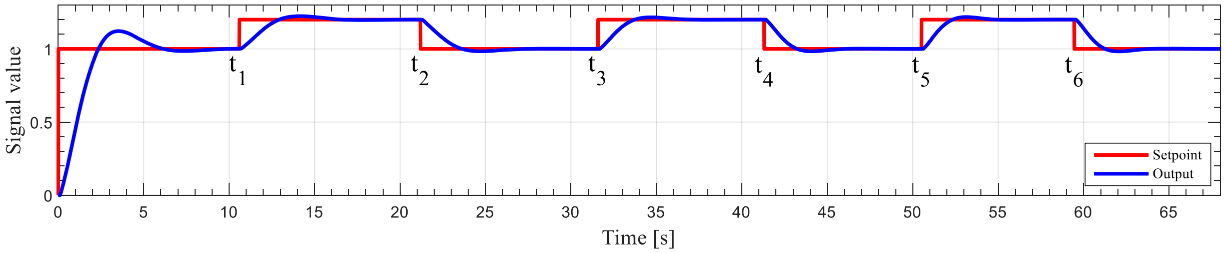

Figure 12. Procedure of tuning PID controller parameters for process G_7 (22) with parameters $T = 1$ and $L = 0.1$ using the proposed FES tuning method in the balanced mode.

Table 9. Overview of PID parameters through iterations of the proposed FES tuning method in the balanced mode for process G_7 (22) with parameters T = 1 and L = 0.1.

Iteration	K_p	T_i	T_d
1	1.24	1.10	0.27
2	1.36	1.28	0.27
3	1.63	1.38	0.25
4	1.88	1.45	0.23
5	2.05	1.49	0.23

and $T = 1.39$. Applying A-H tuning, the following PID parameters are obtained: $K_p = 1.24$, $T_i = 1.1$, and $T_d = 0.27$. The FES tuning method is applied in the balanced mode, and the procedure through iterations can be seen in Figure 12. The procedure starts at moment t_1 and ends at moment t_6. After five iterations, the following PID parameters are obtained: $K_p = 2.05$, $T_i = 1.49$, and $T_d = 0.23$. Table 9 shows how parameters are changed through iterations, from the first one to the last one. For all five iterations, the responses of the system to step changes in setpoint and load are shown in Figure 13. The figure shows that overshoot, rise time, and settling time decrease throughout iterations. It can also be seen that the response to the unit step changes in load improves.

Based on the performance and robustness measures shown in Table 10, it can be concluded that the proposed method meets the previously set goals.

The procedure is repeated for all processes, and improvements are calculated. The average improvements for all groups of processes are displayed in Table 11. The FES tuning method in the balanced mode achieved a significant average improvement of over 30% in settling time and overshoot, which is the main goal of this mode. At the same time, the average improvement in the settling time and integral of the absolute error of the reference and load step response is significant with approximately equal improvements across groups. A degradation of sensitivity is visible, with a highly satisfactory value of 12% on average. Considering all the processes, the average duration of the tuning procedure is about seven iterations.

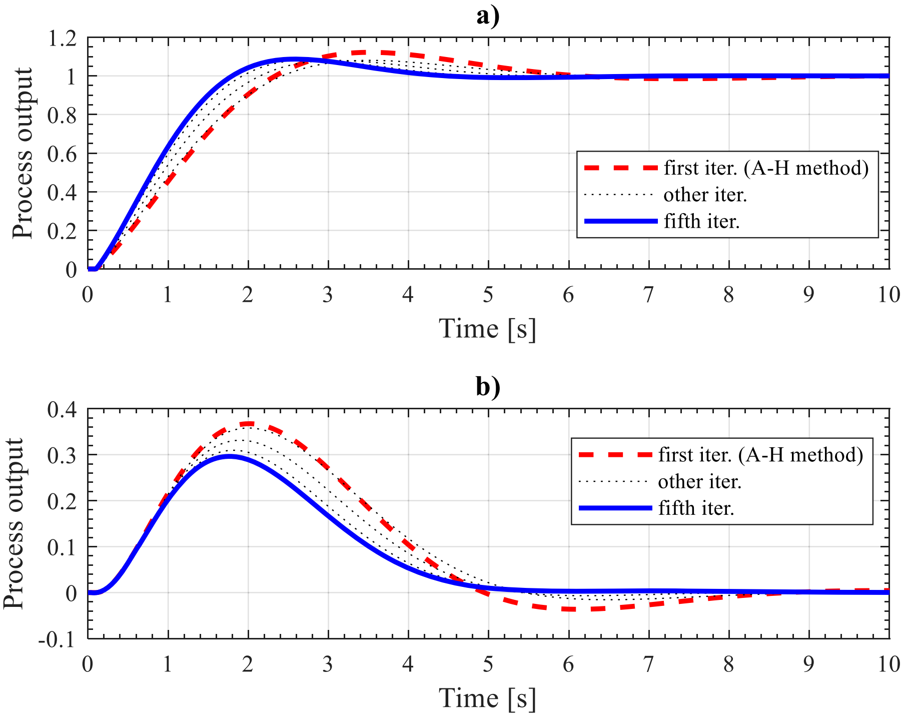

Figure 13. Response to a unit step change in setpoint (a) and load (b) for process G_7 (22) with parameters $T = 1$ and $L = 0.1$. The PID controller is tuned by the A-H method and then fine-tuned by the proposed FES tuning method in the balanced mode.

Table 10. Performance and robustness of the PID controller for process G_7 (22) with parameters $T = 1$ and $L = 0.1$ tuned by the A-H method and fine-tuned by the FES tuning method in the balanced mode.

	t_r [s]	o [%]	t_s [s]	IAE	IAE_d	M_s
A-H method	1.65	12.21	5.60	1.43	1.05	1.27
FES tuning – balanced mode	1.23	8.71	3.94	0.99	0.73	1.25
Improvement [%]	−25.46	−28.65	−29.67	−30.36	−30.35	−1.08

Table 11. Average improvement of performance and robustness measures for all processes achieved by the FES tuning method in the balanced mode after A-H initial tuning.

Process type	Count	t_r [%]	o[%]	t_s [%]	IAE [%]	IAE_d [%]	M_s [%]	Iterations
G_1 (16)	21	−37.81	−12.67	3.02	−17.04	−8.57	28.23	12.19
G_2 (17)	20	−38.51	−13.07	−16.99	−23.78	−23.12	17.53	13.70
G_3 (18)	10	−26.03	−35.00	−20.57	−28.66	−20.26	3.02	5.40
G_4 (19)	5	−28.23	−82.89	−15.56	−20.06	−17.75	9.75	4.60
G_5 (20)	8	−26.18	−30.72	−34.57	−26.98	−25.13	5.06	5.00
G_6 (21)	8	−27.13	−32.29	9.00	−20.67	−2.49	5.71	4.25
G_7 (22)	36	−30.17	−43.41	−25.21	−28.94	−22.29	6.05	4.69
G_8 (23)	11	−29.02	−80.88	−28.73	−22.46	−19.76	12.07	4.45
G_9 (24)	10	−25.91	−37.03	−16.44	−25.75	−19.37	8.44	6.40
All	129	−31.45	−35.80	−16.68	−24.40	−18.36	11.97	7.47

5. Conclusions

This paper studied a novel fuzzy expert scheme for PID auto-tuning. The proposed scheme utilizes the power of the fuzzy logic concept to enhance the tuning of the PID controller. This approach incorporates the control engineer's experience into a fuzzy expert system that would take over the tuning and deliver improved performance over the PID controller tuned using the classical tuning method. The proposed scheme updates the PID controller parameters through iterations based on response characteristics. The tuning process is guided by the scheme that analyzes process data much faster than a human, which results in a significant acceleration of manual tuning.

The scheme has been tested through simulations on a broad set of different processes. The extensive set of processes was used to emphasize the power of generalization of the proposed solution. The A-H method as a representative of classical tuning methods was used as a reference point for comparison of obtained results. The A-H method already gives a good result for all observed processes, which leaves less room for improvement. However, the results showed that the proposed scheme could produce improvements in performance and at the same time keep the robustness within reasonable ranges. The operating modes raise the usability of the proposed scheme to a higher level by allowing it to prioritize the goals of the tuning procedure. The results demonstrated that all three operating modes achieved performance improvement following the initial set priorities (speeding up, reduce overshot, or balanced) in a relatively short time with little degradation of robustness.

Future work will focus on applying the proposed scheme to real controllers used in the industrial and building automation application, and verification of the scheme in a real environment.

Acknowledgment

The authors would also like to express their sincere gratitude to prof. Milan Rapaić from University of Novi Sad, Faculty of Technical Sciences and prof. Veljko Milutinović from the University of Belgrade, School of Electrical Engineering for the constructive advice that has contributed to the quality of this paper.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The authors are grateful for the support of the Ministry of Education, Science and Technological Development of the Republic of Serbia, grant number TR32018.

ORCID

Ilija Kamenko http://orcid.org/0000-0003-3352-7637
Velimir Čongradac http://orcid.org/0000-0002-2246-8945
Filip Kulić http://orcid.org/0000-0002-0982-3017

References

[1] Ziegler JG, Nichols NB. Optimum settings for automatic controllers. Trans Am Soc Mech Eng. 1942;64:759–768.

[2] Chien KL, Hrones JA, Reswick JB. On the automatic control of generalised passive systems. Trans Am Soc Mech Eng. 1952;74:175–185.

[3] Cohen GH, Coon GA. Theoretical considerations of retarded control. Trans Am Soc Mech Eng. 1953;75(1):827–834.

[4] Hägglund T, Åström KJ. Revisiting the Ziegler-Nichols tuning rules for PI control. Asian J Control. 2002;4(4):364–380. doi:10.1111/j.1934-6093.2002.tb00076.x.

[5] Åström KJ, Hägglund T. Revisiting the Ziegler–Nichols step response method for PID control. J Process Control. 2004;14(6):635–650. doi:10.1016/j.jprocont.2004.01.002.

[6] Šekara TB, Mataušek MR. Revisiting the Ziegler–Nichols process dynamics characterization. J Process

Control. 2010;20(3):360–363. doi:10.1016/j.jprocont.2009.08.004.

[7] Åström KJ, Hägglund T. Advanced PID control. Research Triangle Park (NC): ISA - Instrumentation, Systems, and Automation Society; 2006.

[8] O'Dwyer A. Handbook of PI and PID controller tuning rules. 3rd ed London: Imperial College Press; 2009.

[9] Åström KJ, Hägglund T. Automatic tuning of simple regulators with specifications on phase and amplitude margins. Automatica (Oxf). 1984;20(5):645–651. doi:10.1016/0005-1098(84)90014-1.

[10] Huang H-P, Lin C-H. A stable on-line self-tuning optimal PID controller for a class of unknown systems. Asian J Control. 2007;9(2):151–162. doi:10.1111/j.1934-6093.2007.tb00318.x.

[11] Dey C, Mudi RK. An improved auto-tuning scheme for PID controllers. ISA Trans. 2009;48(4):396–409. doi:10.1016/j.isatra.2009.07.002.

[12] Romero JA, Sanchis R, Balaguer P. Pi and PID autotuning procedure based on simplified single parameter optimization. J Process Control. 2011;21(6):840–851. doi:10.1016/j.jprocont.2011.04.003.

[13] Pavković D, Polak S, Zorc D. Pid controller autotuning based on process step response and damping optimum criterion. ISA Trans. 2014;53(1):85–96. doi:10.1016/j.isatra.2013.08.011.

[14] Matić D, Kamenko I, Nikolić P, et al. Optimal control of fruit and vegetables drying process. J. Process. Energy Agric. 2014;18(3).

[15] Cetin M, Iplikci S. A novel auto-tuning PID control mechanism for nonlinear systems. ISA Trans. 2015;58:292–308. doi:10.1016/j.isatra.2015.05.017.

[16] Kumar R, Singla SK, Chopra V. Comparison among some well known control schemes with different tuning methods. J. Appl. Res. Technol. 2015;13(3):409–415. doi:10.1016/j.jart.2015.07.007.

[17] Berner J, Soltesz K, Hägglund T, et al. An experimental comparison of PID autotuners. Control Eng Pract. 2018;73:124–133. doi:10.1016/j.conengprac.2018.01.006.

[18] Zadeh LA. Fuzzy sets. Inf Control. 1965;8(3):338–353..

[19] Mamdani EH. Application of fuzzy algorithms for control of simple dynamic plant. Proc Inst Electr Eng. 1974;121(12):1585–1588. doi:10.1049/piee.1974.0328.

[20] Čongradac V, Prica M, Paspalj M, et al. Algorithm for blinds control based on the optimization of blind tilt angle using a genetic algorithm and fuzzy logic. Sol Energy. 2012;86(9):2762–2770. doi:10.1016/j.solener.2012.06.016.

[21] Zdešar A, Cerman O, Dovžan D, et al. Fuzzy control of a helio-crane: Comparison of two control approaches. J Intell Robot Syst Theory Appl. 2013;72(3–4):497–515. doi:10.1007/s10846-012-9796-0.

[22] Bačkalić T, Bugarski V, Kulić F, et al. Adaptable fuzzy expert system for ship lock control support. J. Navig. 2016;69(6):1341–1356. doi:10.1017/S0373463316000242.

[23] Blažič S, Zdešar A. An implementation of an evolving fuzzy controller. 2017 evolving and Adaptive Intelligent Systems (EAIS); 2017, pp. 1–7, doi:10.1109/EAIS.2017.7954830.

[24] Gao Y, Xiao F, Liu J, et al. Distributed soft fault detection for interval type-2 fuzzy-model-based stochastic systems with wireless sensor networks. IEEE Trans Ind Informatics. 2019;15(1):334–347. doi:10.1109/TII.2018.2812771.

[25] Devi Vidhya S, Balaji M. Hybrid fuzzy PI controlled multi-input DC/DC converter for electric vehicle application. Automatika. 2020;61(1):79–91. doi:10.1080/00051144.2019.1684038.

[26] Du P, Pan Y, Li H, et al. Nonsingular finite-time event-triggered fuzzy control for large-scale nonlinear systems. IEEE Trans. Fuzzy Syst. 2020: 1. doi:10.1109/TFUZZ.2020.2992632.

[27] Farsizadeh H, Gheisarnejad M, Mosayebi M, et al. An intelligent and fast controller for DC/DC converter feeding CPL in a DC microgrid. IEEE Trans Circuits Syst II Express Briefs. 2020;67(6):1104–1108. doi:10.1109/TCSII.2019.2928814.

[28] Nguyen AT, Taniguchi T, Eciolaza L, et al. Fuzzy control systems: past, present and future. IEEE Comput Intell Mag. 2019;14(1):56–68. doi:10.1109/MCI.2018.2881644.

[29] Tzafestas S, Papanikolopoulos NP. Incremental fuzzy expert PID control. IEEE Trans Ind Electron. 1990;37(5):365–371. doi:10.1109/41.103431.

[30] Jigang H, Jie W, Hui F. An anti-windup self-tuning fuzzy pid controller for speed control of brushless dc motor. Automatika. 2017;58(3):321–335. doi:10.1080/00051144.2018.1423724.

[31] He SZ, Tan S, Xu FL, et al. Fuzzy self-tuning of PID controllers. Fuzzy Sets Syst. 1993;56(1):37–46. doi:10.1016/0165-0114(93)90183-I.

[32] Zhao ZY, Tomizuka M, Isaka S. Fuzzy gain scheduling of PID controllers. IEEE Trans Syst Man Cybern. 1993;23(5):1392–1398. doi:10.1109/21.260670.

[33] Petrov M, Proychev T, Topalov A. Expert PID controller with fuzzy self-tuning. IFAC Proc. 1995;28(5):367–372. doi:10.1016/S1474-6670(17)47254-5.

[34] Almeida OdM, Coelho AAR. A fuzzy logic method for autotuning a PID controller: SISO and MIMO systems. 15th triennial World Congress; 2002, pp. 1–6.

[35] Niewiara Ł, Zawirski K. Auto-tuning of PID controller based on fuzzy logic. Comput Appl Electr Eng. 2013;11:230–240.

[36] Wang Y, Jin Q, Zhang R. Improved fuzzy PID controller design using predictive functional control structure. ISA Trans. 2017;71:354–363. doi:10.1016/j.isatra.2017.09.005.

[37] Sivalingam R, Chinnamuthu S, Dash SS. A modified whale optimization algorithm-based adaptive fuzzy logic PID controller for load frequency control of autonomous power generation systems. Automatika. 2017;58(4):410–421. doi:10.1080/00051144.2018.1465688.

[38] Khooban MH. Hardware-in-the-loop simulation for the analyzing of smart speed control in highly nonlinear hybrid electric vehicle. Trans Inst Meas Control. 2019;41(2):458–467. doi:10.1177/0142331218764784.

[39] Gheisarnejad M, Khooban MH. Design an optimal fuzzy fractional proportional integral derivative controller with derivative filter for load frequency control in power systems. Trans Inst Meas Control. 2019;41(9):2563–2581. doi:10.1177/0142331218804309.

[40] Visioli A. Comparison of fuzzy logic based tuning methods for PID controllers. European Control Conference, ECC 1999 - Conference Proceedings; 1999, pp. 497–502, doi:10.23919/ecc.1999.7099353.