

Article

Intelligent Controller Design by the Artificial Intelligence Methods

Jana Nowaková ^{1,*}  and Miroslav Pokorný ² 

¹ Department of Computer Science, Faculty of Electrical Engineering and Computer Science, VSB-Technical University of Ostrava, 17. listopadu 2172/15, 708 33 Ostrava – Poruba, Czech Republic

² Department of Cybernetics and Biomedical Engineering, Faculty of Electrical Engineering and Computer Science, VSB-Technical University of Ostrava, 17. listopadu 2172/15, 708 33 Ostrava – Poruba, Czech Republic; miroslav.pokorny@vsb.cz

* Correspondence: jana.nowakova@vsb.cz; Tel.: +420-597-325-908

Received: 31 May 2020; Accepted: 8 August 2020; Published: 10 August 2020



Abstract: With the rapid growth of sensor networks and the enormous, fast-growing volumes of data collected from these sensors, there is a question relating to the way it will be used, and not only collected and analyzed. The data from these sensors are traditionally used for controlling and influencing the states and processes. Standard controllers are available and successfully implemented. However, with the data-driven era we are facing nowadays, there is an opportunity to use controllers, which can include much information, elusive for common controllers. Our goal is to propose a design of an intelligent controller—a conventional controller, but with a non-conventional method of designing its parameters using approaches of artificial intelligence combining fuzzy and genetics methods. Intelligent adaptation of parameters of the control system is performed using data from the sensors measured in the controlled process. All parts designed are based on non-conventional methods and are verified by simulations. The identification of the system's parameters is based on parameter optimization by means of its difference equation using genetic algorithms. The continuous monitoring of the quality control process and the design of the controller parameters are conducted using a fuzzy expert system of the Mamdani type, or the Takagi–Sugeno type. The concept of the intelligent control system is open and easily expandable.

Keywords: intelligent controller; PID controller; artificial intelligence; expert systems; fuzzy methods; genetic algorithms; optimization; softcomputing

1. Introduction

Sensor-based measuring systems are used not only in the information systems but also in the control ones. Quality control of real complex processes requires continuous monitoring of the changing internal and external conditions, followed by continuous adaptation of the controller parameters. Nowadays, sensor networks are widely used and easily available. We are facing a situation that the sensors provide us with even more information that we are able to use. We can work with a really massive amount of data—big data is trying to find hidden context in data, and the next step is to use the information contained in the data. The world of processes controlling should also change. It must reflect the situation of the enormous amount of data obtained from the processes as well as being able to work with it. The main idea of the work is to present a non-conventional method of solving classic control problems—how to set the controller parameters and when and how to reset them.

The controlling system design for continuous adaptation under time changing conditions is often discussed in many application issues.

Up to the present time, many methods and controlling schemes have been developed. There is a large group of classic design methods [1], techniques based on artificial intelligence [2], or ideas built on a combination of traditional and non-conventional methods. The combination of the methods exploits the best both of these methods provide. The fact that the use of non-conventional methods offers the possibility to add inputs and rules, which are elusive for the standard methods, is one of the biggest advantages.

The proposed solution of the controlling system combines continuous adaptation of the PID controller, monitoring of control process quality according to the predetermined requirements in real conditions of dynamically changing attributes of a controlled system, and ongoing identification of the controlled system from the sensor data. The sensitive and accurate sensors and memory for the data measured form the necessary parts of the system.

1.1. Related Work

Process controlling is a broad scientific field involving various areas. The essential areas of interest comprise intelligent controlling including its parts—control quality monitoring, system identification, and controller parameter adjustment, which are designed using artificial intelligence techniques. The perspective may be formulated as the use of expert systems in controlling and the use of non-conventional optimization methods in controlling. These two groups were developed simultaneously, and several mixed (hybrid) algorithms are also proposed.

As stated in [2], there are two approaches in controlling using an expert system (ES). The first one is the fuzzy rule-based method for controlling processes for which suitable models do not exist or are inadequate. The rules substitute for conventional control algorithms. The second method, originally suggested in [3,4], involves the use of an ES to extend the amounts of classic control algorithms.

It was 1984 when Moore et al. [5] introduced the use of an ES in industrial applications. The system was designed to monitor a process loop of a conventional control system with an intelligent alarm system.

In 1986 Åström, Anton and, at the time Åström's Ph.D. student, Ärzén [4], presented an idea to replace heuristic logic with an expert system in the feedback loop in a classic controlling issue. This has led to a simplification of heuristic logic and has brought new functionalities usable in control systems. A classic PID is considered as the controller. They propose some ideas on how to design a monitoring block and PID controller parameters. This proposed approach can be regarded as the first mention and a basis for expert control, where all the parts are knowledge-based.

Shirley [6] describes the experience with the use of the ES in controlling, namely in controlling an adipic acid reactor and a heat exchanger. The limits of the ES, which should not be considered, according to the author, as a panacea, have been mentioned. From the author's point of view, expert systems are not experts; it means that the ES does not have all the data that an expert could have. Currently, great attention is paid to the ES. The ES should be chosen according to the specific application. The costs of the ES are declining and acquiring knowledge from experts should be conducted with care, with the consciousness that computer power and memory are finite. General rules could be more powerful than more specific rules, and there is a necessity of "good" data. The essence of the article is that Shirley was sceptical about using the ES.

Nowadays, 30 years later, the situation has changed. Based on our work, it can be asserted that the issue of intelligent controllers has been experiencing a renaissance in the last three or four years. One group of PID controller parameters optimization and setting approach uses ant colonies, genetic algorithms, and artificial neural networks [7]. The procedure, which the authors also call intelligent controlling, involves using a single adaptive neuron—direct neural control [8] with a hybrid learning based on the conjunction of an on-line rule based on the gradient descent method and an off-line rule based on the Big Bang-Big Crunch algorithm. A two-layer feed-forward neural network with as few neurons as possible with two inputs, e and Δe , is also proposed. The output is in the form of a change in the duty cycle for the motor control [9].

Non-conventional approaches have become established in industrial applications in the last few years. Controlling of a hydraulically driven robot based on the implementation of two schemes is presented in [10]. The first scheme is created using a fuzzy-PID self-tuning controller composed of the conventional PID control and including fuzzy logic. The second one consists of the adaptive neuro-fuzzy inference system-PID (ANFIS-PID), self-tuning of the gains of the PID controller. However, the fuzzy logic controller [11] or the smart method of controlling [12] (and many others) are very often confused for the intelligent controller.

Conventional PID controller parameters can also be set [13] by the ES of the Mamdani type with two inputs— e and Δe and three outputs— Δk_p , Δk_i and Δk_d . The knowledge base consists of 49 easy rules for each output. An almost identical approach was selected for the PID controller in the photovoltaic system [14]. The other one, the Sugeno type system, extends it, and it is used for optimization and shape definition of input variables membership function. Next, another Sugeno type of the ES for PID controller parameters determination is also proposed.

An effortless and user-friendly method of using decision-based controlling for Smart Home is proposed in [15]. A combination of an artificial neural network and the fuzzy logic controlling system can be devised for a switched reluctance generator, but the controlling systems work independently, without any connection [16].

An exciting implementation of Brain Emotional Learning Based Intelligent Controller is described in [17]. It is proposed for the coordination of multi-agent systems and based on emotional learning in the mammalian limbic system.

Bigdeli and Ziazi [18] designed a robust fractional-order adaptive intelligent controller devised for the stabilization of uncertain fractional-order chaotic systems. The authors prepare the intelligent neuro-fuzzy network for unknown dynamics estimation of the system, while the neuro-fuzzy network parameters, upper bounds of the model uncertainties, disturbances, and approximation errors are adaptively estimated via separate adaptive rules.

The authors of [19] set the goal to improve the control performance for the uncertain nonlinear systems. The idea of the study is to use a self-organizing, fuzzy neural network to imitate the control law directly and, then, to appeal to obtain a compact structure of the controller to further reduce the computational burden and to enhance the control performance.

Controlling system tuning by an online fuzzy-based approach was proposed to intelligently optimize the gain values in the load frequency control structure in [20]. Then the intelligent fuzzy based inference system for autonomous mobile robot navigation is presented by [21], for a rigid robotic manipulator system, it is presented by [22] or, for bicycle robot balancing, it is presented by [23].

System identification is also a necessary task for the design of the appropriate controlling system. This is an estimation of the system parameters from the data observed. There are a lot of classic system parameters identification methods, such as nonparametric time- and frequency-domain or estimation methods [24,25]. In the identification area, some problems can be formulated as optimization tasks. They can be solved by various high-level optimization strategies employed to minimize (or maximize) an objective function representing the problem. Optimization algorithms can be, in general, divided into two main groups. The first group contains deterministic algorithms, while the second one comprises stochastic (randomized) methods. In our case, the stochastic method—genetic algorithms [26]—is used. Unfortunately, no mention in connection with the use of genetic algorithms in process identification can be found. On the contrary, references to the use of fuzzy clustering [27] or neural networks [28,29] can be found.

1.2. The Features of the Proposed Solution

The proposed solution is based on soft-computing techniques, is open, flexible, and provides the possibility of expert knowledge embedding. The final intelligent controller is modular. It comprises a module of ongoing process quality monitoring, a module for processing transfer function identification and a module for setting the PID controller parameters. All the modules could be used separately as

well as in other concepts for process control. The modules are designed generally for controlling a system of a higher order.

The algorithms of the artificial intelligence procedure are formally as simple as possible, understandable for the user and usable in the area of education of new engineers for their clarity. This type of solution minimizes the requirements for computing performance and is suitable for the use in embedded systems. The simulation experiments are selected to prove the efficiency of the non-conventional methods in comparison with the conventional ones.

The main idea is to effectively combine the non-conventional methods with the classic ones and to present the possible use. Such a solution allows the use of a combination of exact measurements and vague expert knowledge to improve monitoring and control procedures. It has been decided to combine two approaches to the artificial intelligence area. The first one is the use of the fuzzy-logic rule-based models (expert systems) for the quality improvement of continuous monitoring of the controlling process from the sensor data. This fuzzy system is of the Mamdani type with rule-based operating under two inputs, but it can be very easily extended [30]. The output of the system is the decision if the actual quality of the controlled process is appropriate, or there is necessity to re-adapt the controlling system. The second fuzzy expert system of the Takagi–Sugeno type is proposed for the controlling system parameters design according to the decision of the controlling process quality [31]. This tool is also open as for all its aspects, and its knowledge base is built on traditional time-tested PID controller design methods. For the identification of the controlled system parameters, the genetics algorithms optimization methods are proposed. Genetics algorithms are used for finding the parameters of the difference equation of the controlled system.

When setting the solution to the state of the art of the intelligent controlling area, the proposed design is sophisticated. All the modules are designed on the basis of unconventional methods. The family of the controlling process is more extensive; it is not designed for one particular controlled system. The paper has not presented a solution for one controlling issue. Yet, it is more general, but it includes the possibility to adapt the modules to a specific issue in a simple manner.

The remaining part of this paper is organized as follows. Section 1 summarizes the work relating to intelligent controlling systems. Section 2 outlines the proposed design and details the principles of all parts of the systems presented. Section 3 provides the numerical verification and the verification by simulation to illustrate the ability of the proposed solution to find an accurate response to the changing conditions of the controlled system. First of all, the modules are presented and simulated separately to show the efficiency of every module in various cases. Then, the entire intelligent controller is simulated with the description of the re-adaptation strategy. Section 4 provides a short discussion on the related topics.

2. Materials and Methods

It has been mentioned in the beginning that the idea is to design an intelligent controlling system, where all the essential parts are built on the use of non-conventional methods. A combination of two approaches from the field of artificial intelligence—fuzzy-logic expert systems for the decision of the appropriate or non-appropriate controlling quality and for controller parameters setting—was proposed. Then, a non-conventional identification of the controlled system parameters by genetic algorithms was designed. All the parts can be used separately or together. The intention was to design systems based on fuzzy-logic principles [32] and soft-computing methods [33,34], as well as novel systems based on standard methods [35]. A simplified system block schema (Figure 1) used three modular blocks—ES1, IS, and ES2. The schema was built on the primary negative feedback loop controlling with two sensors placed for measuring the input $u(t)$ and output $y(t)$ of the controlled system. All the parts are going to be described in detail. The design is proposed more generally, but provides a particular example.

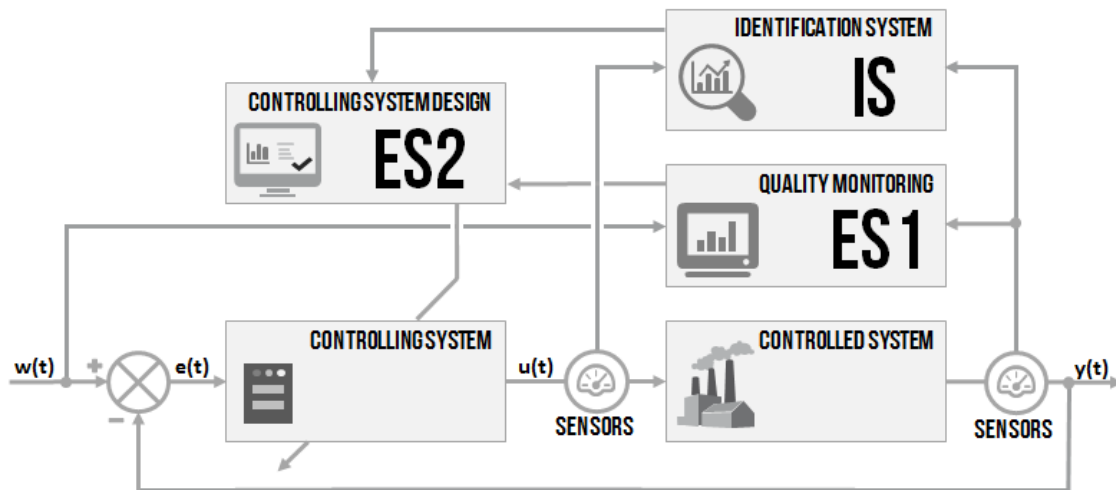


Figure 1. The schema of the proposed intelligent controlling system.

2.1. Quality Monitoring–ES1

This section focuses on the adjustment of the quality monitoring system (ES1) [30] for deciding when to re-adapt the classic PID controller. In the schema of the intelligent controller, it was connected to a block for controlling the system design. The monitoring system presented has some common elements with [36]. A fuzzy expert system [37] of the Mamdani type [32,38] was created. This system contains two inputs (relative overshoot and settling time), but the settling time is not defined as a classic time quantity, but as the part or multiple of the previously measured settling time (relative settling time), and one output (score). The score determines if it is necessary to re-adapt the controller.

To better understand the article and the issue itself, a block containing monitoring procedure on the schema of the intelligent controller is outlined in Figure 2.

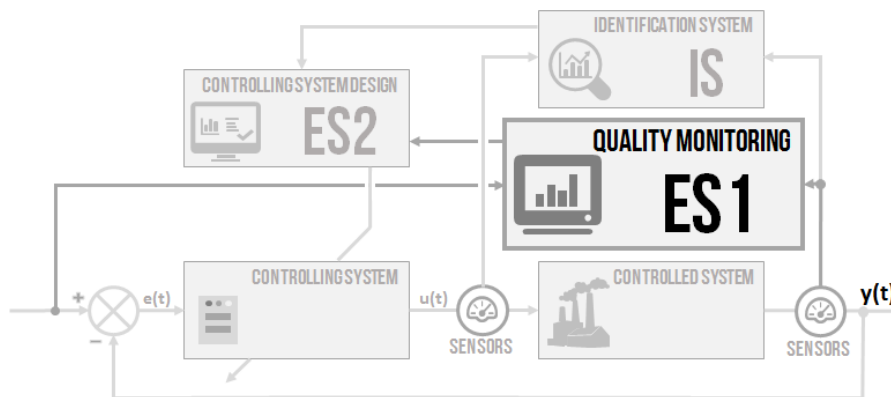


Figure 2. The schema of the concept of intelligent controlling system with outlined block of monitoring procedure.

2.1.1. Quality Monitoring Inputs

The fuzzy expert system ES1 [30] is a decision support system [39] with two inputs, which are easily obtained from timing a controlled process—relative overshoot and relative settling time. The meaning of these variables is depicted in (Figure 3). As the relative overshoot, the first peak overshoot rated to the steady-state value of the step response was chosen. The settling time was the time from which the timing stays in the range of $\pm p\%$ of the steady-state value of the system step response [1].

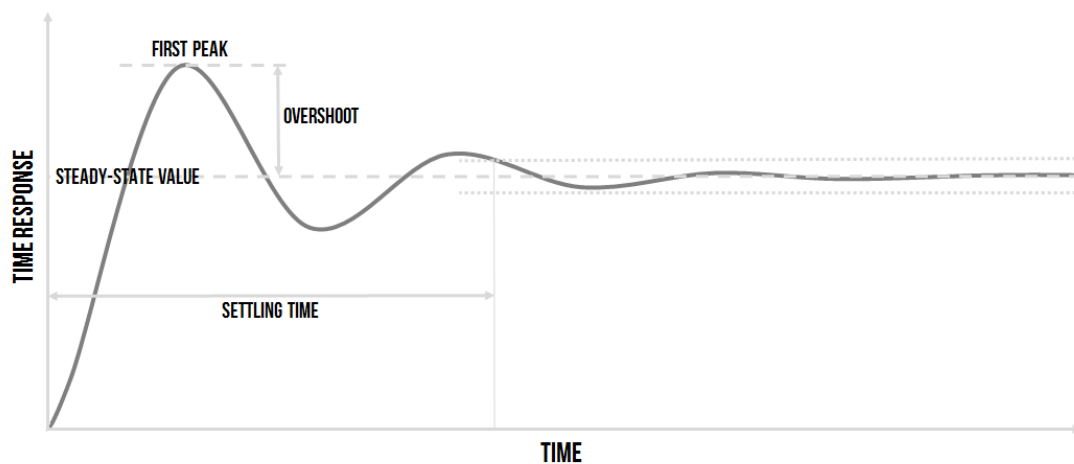


Figure 3. The depiction of the overshoot and the settling time in general timing.

ES1 was the fuzzy expert system, so the inputs were the linguistic variables with the following meaning. The first input was called Relative Settling Time (RST), it was the linguistic variable with settable linguistic values. The Relative Settling Time (Equation (1)) was computed as the ratio of the actual settling time (ST_k) and the settling time of previous steady state (ST_{k-1}):

$$RST_k = \frac{ST_k}{ST_{k-1}}. \quad (1)$$

It is necessary to save the settling time of the previous steady-state in the memory. The settling time is often set as a 2% standard deviation from the steady-state value [1], but the user can choose its value in our opinion, it is an optional value.

It is proposed to set the linguistic values of the RST as follows (Figure 4):

- Faster—the settling time, in comparison with the previous settling time, is shorter
- Optimal—the settling time and the previous settling time are almost the same
- Slower—the settling time, in comparison with the previous settling time, is longer

The numerical description of the broken-line shape of the linguistic values membership functions could be provided as follows:

- Faster— $[F_1 \ F_2 \ F_3]$, where e.g., $F_1 = 0$, $F_2 = 0$ and $F_3 = 1$, it means Faster— $[0 \ 0 \ 1]$, the right border $F_3 = 1$ is equal to the idea that, in this point, the RST of previous and actual time course are the same and do not belong to this linguistic Faster value.
- Optimal— $[O_1 \ O_2 \ O_3]$, where e.g., $O_1 = 0$, $O_2 = 1$ and $O_3 = 2$, it means Optimal— $[0 \ 1 \ 2]$. The Optimal value means that the actual settling time is around the value of the previous settling time and is not greater than two times. The right border $O_3 = 2$ could be set as another value, in this case the monitoring system is not going to be so “strict”.
- Slower— $[S_1 \ S_2 \ S_3 \ S_4]$, where e.g., $S_1 = 1$, $S_2 = 2$, $S_3 = 100$ and $S_4 = 100$, it means Slower— $[1 \ 2 \ 100 \ 100]$. The value of S_2 can also be set to the other value according to the value of O_3 ; the bigger S_2 and O_3 (proposing $S_2 = O_3$) are the less strict the monitoring system is. The values exceeding the limit of S_4 are considered as equal to S_4 , as the limit of the linguistic Slower value.

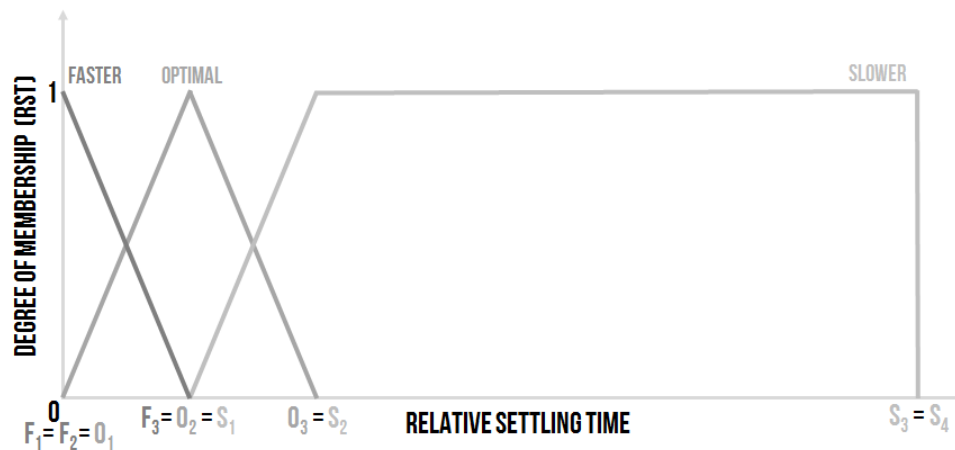


Figure 4. The shape of the membership functions of the linguistic values for the input linguistic variable Relative Settling Time (RST) [30].

The linguistic input called Relative Overshoot (RO) was selected as the second input of the fuzzy expert system for the quality of the time response. The value of the overshoot can be seen in the time response of the controlled system, and it is detected after any change in the closed controlling loop. The relative overshoot is represented as the difference between the maximal overshoot (MO) of the controlled variable ($y(t)$) and the change of the setpoint—the required value ($w(t)$) (see Figure 1) is rated relative to the required value (Equation (2)):

$$RO = \frac{|MO - w(t)|}{|w(t)|}. \quad (2)$$

The relative overshoot is expressed in percentage. The value of the Relative Overshoot should be stored in the memory along with the value of the Relative Settling Time.

This linguistic input was settable, and the use of three linguistic values was proposed (Figure 5):

- Low—the relative overshoot is small, and, from the controlling point of view, it is a good state.
- Appropriate—the value of the relative overshoot is small enough, the border for the acceptable overshoot is often set as 20%.
- High—the relative overshoot is high, which is not good for many controlled systems and especially for the sensors (which are used for the measurement of the required value). The influence of a high overshoot on the sensors could be fatal. The sensors can be destroyed by a high overshoot, which is a better option. Or, the high overshoot can damage the sensor, but the sensor will function, although inaccurately. The poor functionality of the sensor will not be clear at first sight, and the entire system will work unpredictably.

The shape of the membership functions of the linguistic values expressed by the fuzzy sets can be written using the numerical entry:

- Low— $[L_1 \ L_2 \ L_3 \ L_4]$, where e.g., $L_1 = 0, L_2 = 0, L_3 = 0.1$ and $L_4 = 0.15$, it means Low— $[0 \ 0 \ 0.1 \ 0.15]$.
- Appropriate— $[A_1 \ A_2 \ A_3]$, where e.g., $A_1 = 0.1, A_2 = 0.15$ and $A_3 = 0.2$, it means Appropriate— $[0.1 \ 0.15 \ 0.2]$.
- High— $[H_1 \ H_2 \ H_3 \ H_4]$, where e.g., $H_1 = 0.15, H_2 = 0.2, H_3 = 10$ and $H_4 = 10$, it means High— $[0.15 \ 0.2 \ 10 \ 10]$.

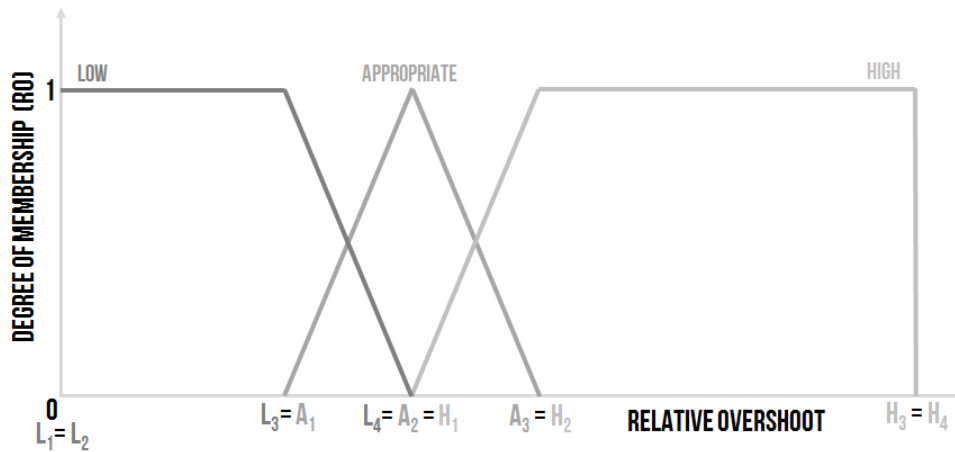


Figure 5. The shape of the membership functions of the linguistic values for the input linguistic variable Relative Overshoot (RO)–design no. 1 [30].

The other possible setting could be defined as follows (Figure 6):

- Low– $[L_1 L_2 L_3]$, where e.g., $L_1 = 0, L_2 = 0$ and $L_3 = 0.15$, it means Low– $[0 0 0.15]$.
- Appropriate– $[A_1 A_2 A_3]$, where e.g., $A_1 = 0, A_2 = 0.15$ and $A_3 = 0.2$, it means Appropriate– $[0 0.15 0.2]$.
- High– $[H_1 H_2 H_3 H_4]$, where e.g., $H_1 = 0.15, H_2 = 0.2, H_3 = 10$ and $H_4 = 10$, it means High– $[0.15 0.2 10 10]$.

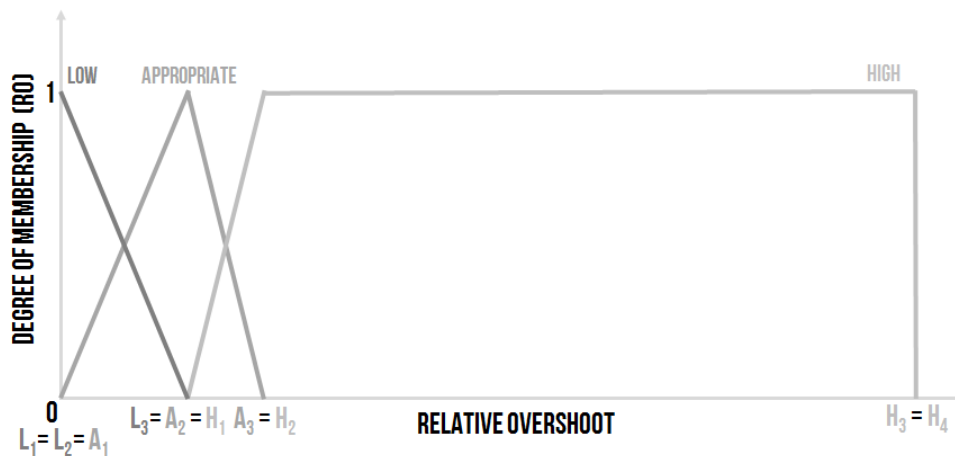


Figure 6. The shape of the membership functions of the linguistic values for the input linguistic variable Relative Overshoot (RO)–design no. 2.

The second design proposal is not so “strict”. The design depends on the expert’s experience, the nature of the controlled system, and the method of the controlling strategy.

2.1.2. Quality Monitoring Output

Expert system ES1 represents a system with two linguistic inputs and one linguistic output–Score. The Score linguistic variable must be defuzzified using the Center of Area method (COA) [33]. The higher the value of the Score is the more the controlling process is considered as good enough and the selected controller as appropriate. It was set that score value of less than “two” is regarded as unsatisfactory, and the controller has to be re-adapted [36]. Everything equalling or exceeding “two” is regarded as good enough.

The Score linguistic variable is also a linguistic variable with linguistic values expressed as fuzzy sets (Figure 7). In this case, four linguistic values were used, and their triangular-shape membership functions could be described as follows:

- Approximately zero— $[0\ 0\ 1]$,
- Small— $[0\ 1\ 2]$,
- Medium— $[1\ 2\ 3]$,
- Large— $[2\ 3\ 3]$.

The linguistic values of the Score can be adjusted for a particular case if necessary.

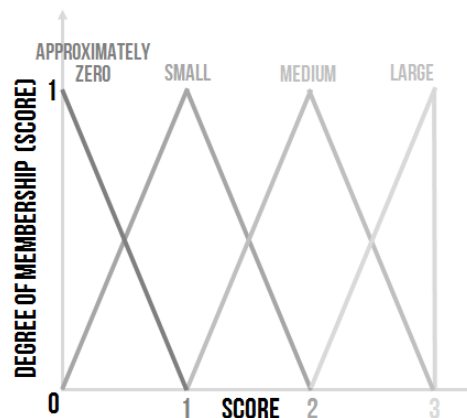


Figure 7. The shape of the membership functions of linguistic values for the output linguistic variable Score [30].

2.1.3. Quality Monitoring System Knowledge Base

The knowledge base was formed by nine linguistic IF-THEN rules of the Mamdani type [33]:

1. If (RO is *Low*) and (RST is *Slower*) then (Score is *Medium*)
2. If (RO is *Low*) and (RST is *Optimal*) then (Score is *Large*)
3. If (RO is *Low*) and (RST is *Faster*) then (Score is *Large*)
4. If (RO is *Appropriate*) and (RST is *Slower*) then (Score is *Small*)
5. If (RO is *Appropriate*) and (RST is *Optimal*) then (Score is *Medium*)
6. If (RO is *Appropriate*) and (RST is *Faster*) then (Score is *Large*)
7. If (RO is *High*) and (RST is *Slower*) then (Score is *Approx. zero*)
8. If (RO is *High*) and (RST is *Optimal*) then (Score is *Small*)
9. If (RO is *High*) and (RST is *Faster*) then (Score is *Medium*)

The shape of the membership function of the output variable was inferred using the Mamdani method [33]. As the inference method, the Fuzzy Modus Ponens derivation rule was implemented [40]. The linkage between the rules in the knowledge base is disjunctive. Fuzzy conjunction and fuzzy implication were interpreted as basic logic functions—fuzzy disjunction, fuzzy conjunction and fuzzy implication were interpreted as maximum, minimum and minimum resp. According to this fact, the algorithm of the derivative methods was straightforward. The crisp value of the Score output was performed using the Center of Area defuzzification method.

2.2. Identification System—IS

Controlled system parameters identification is a known and necessary task in controlling the design and its following adaptation [41]. System identification is used to assess the nature of the systems. The use of genetic algorithms, as the optimization procedure for finding the parameters of the system transfer function, is proposed. The system representation in the form of the transfer

function was selected because it was available from the sensor data. It was suitable for a black-box description, and it was easy to use for our next step—controlling the system adaptation, if necessary. An identification system (IS) is generally designed to be connected to ES2 in the intelligent controller schema. The information on the ongoing identification is necessary for the re-adaptation of the PID controller parameters. The method presented can be used both for continuous and discrete systems [42,43].

The idea of the system prepared was to identify the parameters of the difference equation from the sensors data—the input–output values of the system measured (system input–control variable, system output). It was built on the optimization feature of genetic algorithms (GA) [44]. In our case, GAs were used to find the coefficients of the difference equation of the controlled system according to the selected fitness function.

The block containing the identification procedure on the schema of the intelligent controlling system is outlined in Figure 8.

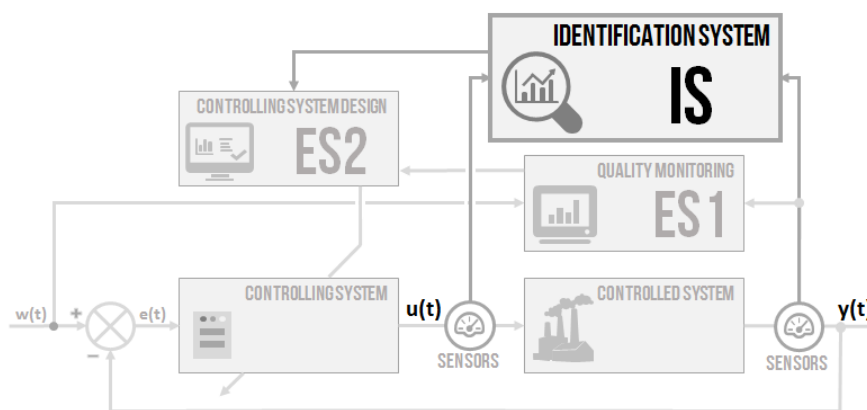


Figure 8. The schema of the concept of intelligent controlling system with outlined block of the controlled system identification procedure.

The measurements (sensor data evaluation) were always performed in some time-points (predefined)—with a preset period (sampling period) T . The fact that the sampling period should be small enough to be able to reconstruct the signal according to the Nyquist–Shannon sampling theorem [45] should be taken into consideration.

The identification aims to fit the mathematical model as well as possible and to eliminate the influence of the noise distortions to the greatest extent [41]. As for the proposed method of identification, the presence of additive noise could not affect the result of the identification severely. In reality, the more variable the input signal is the better results could be obtained. Nevertheless, the noise level should not be too high; it means that the signal-to-noise ratio should be big enough.

The identification procedure could be used for any system for which the transfer function can be written. At the beginning, the expert should define the system order according to the nature of the controlled system. Hence, the expert should determine, for example, that the system is of the third order or the system can be identified as a system of, for example, up to the fourth order with the decision according to the accuracy needed. The effectiveness of the identification procedure for the system of the third order is going to be described in Section 3, but we suggest that the second order is sufficient because ES2 is designed for the system of the second order. The only thing which should be borne in mind is that the higher the order of the system is found the longer the computing time/greater computing power is required. Therefore, in general, the system order is not limiting; the only limit is the required time and the computational equipment. As for the second-order system, four unknown values must be found; as for the third-order system, six values must be found; and, as for the fourth-order system, eight values must be found, etc.

Genetic algorithms (GAs) have been one of the first and most popular evolutionary algorithms. GAs are based on computer implementation of biological population-based evolution with selection, mutation, and crossover principles [44].

The third-order system was selected as an example of the use and it can be defined using the transfer function in the following form:

$$G_S(s) = \frac{b_3s^3 + b_2s^2 + b_1s + b_0}{a_3s^3 + a_2s^2 + a_1s + a_0}. \quad (3)$$

The difference equation is then defined as:

$$y(k) = -\frac{a_{k-1}}{a_k}y(k-1) - \frac{a_{k-2}}{a_k}y(k-2) - \frac{a_{k-3}}{a_k}y(k-3) + \frac{b_{k-1}}{a_k}u(k-1) + \frac{b_{k-2}}{a_k}u(k-2) + \frac{b_{k-3}}{a_k}u(k-3), \quad (4)$$

with

$$\begin{aligned} Y_1 &= \frac{a_{k-1}}{a_k}, \\ Y_2 &= \frac{a_{k-2}}{a_k}, \\ Y_3 &= \frac{a_{k-3}}{a_k}, \\ U_1 &= \frac{b_{k-1}}{a_k}, \\ U_2 &= \frac{b_{k-2}}{a_k}, \\ U_3 &= \frac{b_{k-3}}{a_k}. \end{aligned} \quad (5)$$

where Y_1, Y_2, Y_3 and U_1, U_2, U_3 are the optimized parameters for which the GA are used.

For finding of the coefficient of difference equation the fitness function has to be defined, for our solution the fitness function is defined as:

$$J = \frac{1}{N} \sum_{j=1}^N [y^{\text{measured}}(x_j) - y^{\text{GA}}(x_j)]^2 \rightarrow 0, \quad (6)$$

where N is the size of the set of input-output data.

2.3. Controlling System Design—ES2

The entire proposed solution would not be complete without the last part—the controller parameters design system (ES2). Three versions of this module were designed, wherein the know-how obtained from the classic PID controller design methods was implemented. The knowledge from the Ziegler–Nichols (ZN) step response method, a combination of Ziegler–Nichols methods (the combination of the frequency response method and the step response Ziegler–Nichols design method) [46] and one of the possible modification of the Ziegler–Nichols step response method—the Chien, Hrones, and Reswick (CHR) setpoint response design method with a 0% overshoot [1] in a variant with a 0% overshoot (for more information see [31]), was applied.

ES2 is a fuzzy expert system based on a model of the Takagi–Sugeno type without the necessity of the defuzzification procedure [37]. The block containing ES2 on the schema of the intelligent controlling system is outlined in Figure 9. ES2 was connected with the IS and ES1. The connection with the IS provided information about the structure and parameters of the controlled system. The relationship

with ES1 provided the information if the re-adaptation of the controller parameters was required or not.

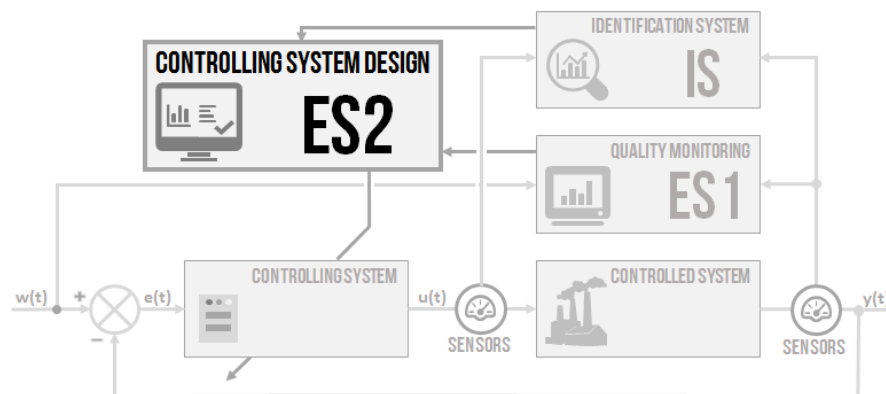


Figure 9. The schema of the concept of intelligent controlling system with outlined block of the controller parameters design system.

2.3.1. Controlling System Design Inputs

The proposal was to diversify the working area of the controlled system to smaller parts and to design, for every part, the appropriate controller using the classic design controller method and the parameters obtained from the PID controller used for the knowledge design. The method should be selected by the expert. A demonstration was conducted for the system of the second order with the following transfer function:

$$G_S(s) = \frac{1}{a_2s^2 + a_1s + a_0}. \quad (7)$$

The constants from the denominator of the transfer function of the controlled system a_2 , a_1 and a_0 were the inputs variables of the ES2 which, as it is further explained, were used for identifying the parameters of a classical PID controller. It was decided that:

- $a_2 \in \langle a_{2,lower}, a_{2,upper} \rangle$,
- $a_1 \in \langle a_{1,lower}, a_{1,upper} \rangle$,
- $a_0 \in \langle a_{0,lower}, a_{0,upper} \rangle$,

with the possible graphical representation in fuzzy sets depicted in Figure 10 or in Figure 11, where, for every linguistic variable, it was proposed to have five linguistic values XSmall (XS), Small (S), Medium (M), Large (L) and XLarge (XL).

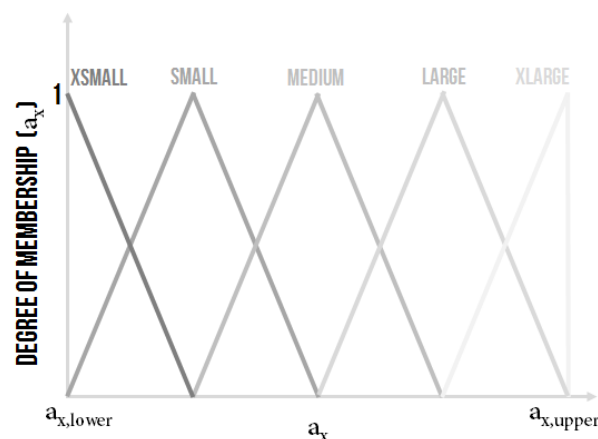


Figure 10. The shape of the membership functions of linguistic values for the input linguistic variable a_2 , a_1 and a_0 .

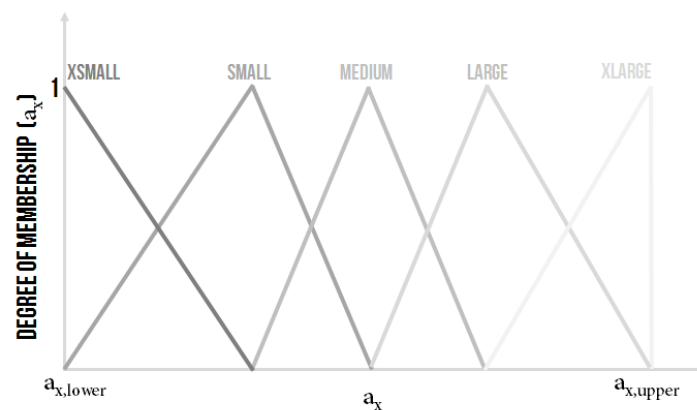


Figure 11. The other possible shape of the membership functions of linguistic values for the input linguistic variable a_2 , a_1 and a_0 .

In the case presented, five linguistic values for every linguistic variable were used, the description of the shape of the linguistic values of the linguistic variable a_2 was as follows:

- XSmall (XS)–[0 0 6],
- Small (S)–[0 6 11],
- Medium (M)–[6 11 16],
- Large (L)–[11 16 22],
- XLarge (XL)–[16 22 22].

The description of the shape of the linguistic values of the linguistic variable a_1 was as follows:

- XSmall (XS)–[0 0 5],
- Small (S)–[0 5 10],
- Medium (M)–[5 10 15],
- Large (L)–[10 15 20],
- XLarge (XL)–[15 20 20].

The description of the shape of the linguistic values of the linguistic variable a_0 was as follows:

- XSmall (XS)–[0 0 9],
- Small (S)–[0 9 14],
- Medium (M)–[9 14 19],
- Large (L)–[14 19 28],
- XLarge (XL)–[19 28 28].

The shape and the range of the linguistic values of the linguistic variable a_2 , a_1 and a_0 were subject to an expert decision.

2.3.2. Controlling System Design Output

The next part focuses on determining the parameters of conventional PID controllers using any appropriate design methods for every combination of the linguistic value of a_2 , a_1 and a_0 . The values where these fuzzy sets have the maximal value of the membership function should be used as the values of a_2 , a_1 and a_0 . For every combination, the crisp values of the constants of K , T_i , and T_d should be calculated.

As it was mentioned before, three versions of ES2 were considered. The first version was based on the Ziegler–Nichols (ZN) step response method, the second one was based on a combination of Ziegler–Nichols methods (the combination of the frequency response method and the step response Ziegler–Nichols design method) [46] and the last one was based on the possible modification of the

Ziegler–Nichols step response method—the Chien, Hrones, and Reswick (CHR) setpoint response design method with a 0% overshoot [1] in a variant with a 0% overshoot (for more information see [31]). These three conventional methods were used to determine the parameters of the conventional PID controller (K , T_i , and T_d) for systems with all combinations of a_2 , a_1 and a_0 . In total, 125 crisp values (a single element fuzzy set) for each output value of K , T_i , and T_d and for each version were created (the entire table was not included, but it could be easily computed).

2.3.3. Controlling System Design Knowledge Base

Using the information mentioned above, the core of the knowledge base was created. The crisp values of $KKNOW$, $TIKNOW$ and $TDKNOW$ represent parameters of the classical PID controller with a transfer function in a form of:

$$G_R(s) = KKNOW \left(1 + \frac{1}{TIKNOW \cdot s} + TDKNOW \cdot s \right). \quad (8)$$

The definition of the linguistic IF-THEN rules of the Takagi–Sugeno type [47,48] is as follows:

R_r : If (x_{a_2} is a_{2r}) and (x_{a_1} is a_{1r}) and (x_{a_0} is a_{0r}) then

then ($KKNOW_r = K_r$) and ($TIKNOW_r = T_{i_r}$) and ($TDKNOW_r = T_{d_r}$),

where $r = 1, 2, \dots, R$ is a number of the rule.

A fragment of the rule may appear as follows:

⋮

10. If (a_2 is M) and (a_1 is S) and (a_0 is S) then ($KKNOW_r = K_{10}$) and ($TIKNOW_r = T_{i_{10}}$) and ($TDKNOW_r = T_{d_{10}}$)
 11. If (a_2 is M) and (a_1 is S) and (a_0 is M) then ($KKNOW_r = K_{11}$) and ($TIKNOW_r = T_{i_{11}}$) and ($TDKNOW_r = T_{d_{11}}$)
 12. If (a_2 is M) and (a_1 is S) and (a_0 is L) then ($KKNOW_r = K_{12}$) and ($TIKNOW_r = T_{i_{12}}$) and ($TDKNOW_r = T_{d_{12}}$)
 13. If (a_2 is M) and (a_1 is S) and (a_0 is XL) then ($KKNOW_r = K_{13}$) and ($TIKNOW_r = T_{i_{13}}$) and ($TDKNOW_r = T_{d_{13}}$)
- ⋮

The inference algorithm of the output value in Takagi–Sugeno models is preformed using the weighted mean value. For the output value of $KKNOW$, it is defined in the following form:

$$KKNOW = \frac{\sum_{r=1}^R \{ [\mu_{a_2}(x_{a_2}^*) * \mu_{a_1}(x_{a_1}^*) * \mu_{a_0}(x_{a_0}^*)] KKNOW_r \}}{\sum_{r=1}^R [\mu_{a_2}(x_{a_2}^*) * \mu_{a_1}(x_{a_1}^*) * \mu_{a_0}(x_{a_0}^*)]}, \quad (9)$$

where μ_{a_2} , μ_{a_1} , and μ_{a_0} represent the degree of membership of the given input (concrete values of the input variable) $x_{a_2}^*$, $x_{a_1}^*$, $x_{a_0}^*$ to its appropriate fuzzy linguistic value in the antecedent of the r^{th} rule. The outputs of the system were represented by the crisp values (constants) of $KKNOW$, $TIKNOW$ and $TDKNOW$ and the defuzzification was not required. The algorithm of the inference method was simple and computationally unpretentious.

For the second-orders controlled system with three input linguistic variables, where every linguistic variable has five linguistic values, it is necessary to define 125 rules (5^3).

3. Results

The efficiency of every module of the proposed intelligent controlling system was demonstrated independently because all the modules could be used independently and can also work independently. Then, the simulation of the use of all the modules connected to the entire intelligent controller together was provided. The modules were presented separately, with different parameters, to demonstrate the scope of the use. The parameters of the given simulation were selected randomly, in the possible range of the modules. The evaluation of the proposed methods and numerical experiments are

presented (simulated) using the Matlab and Simulink software environment, and the identification procedure was also implemented using the C++ programming language. The Matlab and Simulink software environment was chosen for its comprehensiveness, since it is considered a standard, well-known and frequently used instrument for engineering simulation in the area of process controlling and measurements.

3.1. Monitoring of Control Process Quality—ES1

At the beginning, the simulation of the monitoring system was performed with a controlled system of third order using the following transfer function [49]:

$$G_{ES1_1}(s) = \frac{1}{5s^3 + 25s^2 + 8s + 3}, \quad (10)$$

with an appropriate PID controller. The timing of the step response of the feedback system is shown in Figure 12 (upper). The settling time of the step response was:

$$ST_{ES1_1} = 18.0444, \quad (11)$$

and relative overshoot:

$$RO_{ES1_1} = 15.8735. \quad (12)$$

During the controlling process, the controlled system transfer function of the third order was supposed to change to the second order with the following transfer function:

$$G_{ES1_2}(s) = \frac{1}{10s^2 + 7s + 2}. \quad (13)$$

The timing of the step response of the feedback loop is shown in Figure 12 (lower). The settling time of the step response now equalled:

$$ST_{ES1_2} = 12.7590, \quad (14)$$

and the relative overshoot:

$$RO_{ES1_2} = 0. \quad (15)$$

The score using ES1 was inferred from the inputs:

$$RST_{ES1} = \frac{12.7590}{18.0444} = 0.707089, \quad (16)$$

and the relative overshoot:

$$RO_{ES1} = 0, \quad (17)$$

and equals:

$$score = 2.66 > 2. \quad (18)$$

The value of the score was higher than 2, so the step response had appropriate timing; re-adaptation of the PID controller was unnecessary.

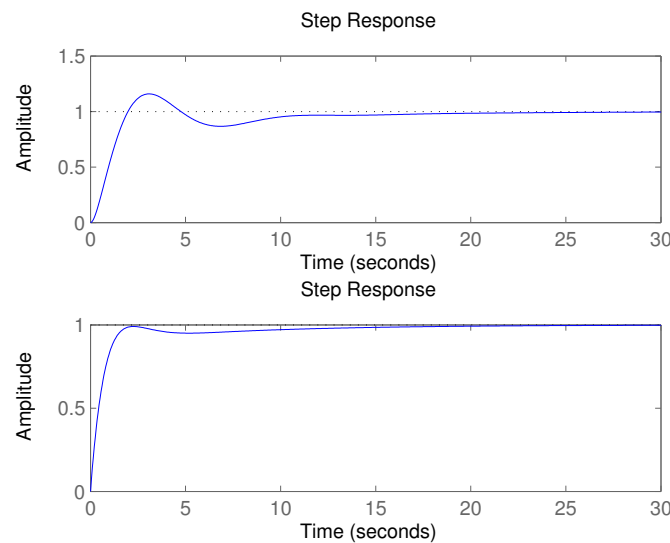


Figure 12. Step response of initial (**upper**) system and changed (**lower**) controlled system according to the simulation of monitoring system [49].

The quality monitoring system was robust enough; the usability was tested for the controlled systems up to the third-order, but there was no reason why it could not work for controlled systems of a higher order. Yet, however, it should be mentioned that stable, controlled systems were taken into consideration here. The next idea stems from the definition of another/other input(s), for which the system was prepared.

3.2. Identification of Transfer Function Parameters–IS

The next part deals with the simulation of the identification module (IS), which is going to be demonstrated for the system of the third-order with the following transfer function [49]:

$$G_{IS}(s) = \frac{s + 1}{s^3 + 3s^2 + 3s + 4}. \quad (19)$$

\mathcal{Z} -transform was used where the sampling period was defined as $T = 0.25$ s; the difference equation of the system was then as follows:

$$y_{original}(k) = -(-2.31816)y(k-1) - 1.83381y(k-2) - (-0.472367)y(k-3) + 0.0264792u(k-1) + 0.00182614u(k-2) - 0.0174828u(k-3). \quad (20)$$

After using the identification procedure, the resulting difference equation was determined as:

$$y_{identified}(k) = -(-2.31758)y(k-1) - 1.83278y(k-2) - (-0.471842)y(k-3) + 0.0264684u(k-1) + 0.00184947u(k-2) - 0.0174791u(k-3), \quad (21)$$

which was practically the same as (20) with the following best value of the fitness function:

$$J = 1.07770 \times 10^{-12}. \quad (22)$$

After applying the inverse \mathcal{Z} -transform with the same $T = 0.25$ s, the continuous transfer function after identification equalled:

$$G_{identified}(s) = \frac{-0.0000539209s^2 + 1.00011s + 1.00199}{s^3 + 3.00444s^2 + 3.00171s + 4.00825}. \quad (23)$$

The approximated system transfer function corresponding to the shape of (20) was determined as:

$$G_{\text{approximated}}(s) = \frac{1.00011s + 1.00199}{s^3 + 3.00444s^2 + 3.00171s + 4.00825} \quad (24)$$

To assess the difference in the step response of the original (19), identified (23) and approximated (24) system, see Figure 13. The error calculated as a definite integral of the absolute value of the difference between the original and identified step responses was 0.2946, or 0.03137 between the original and approximated ones. The graphical representation of the step responses of the original, identified and approximated systems was practically the same as the difference was minimal and the curves merged. The computing time for the experiment presented was 6.386 sec based on the data from 97 samples of the input-output values (the simulation time was chosen as 25 s, using sampling period $T = 0.25$ s; three samples were not used as they were incomplete for the system of the third-order because it was necessary to know the data three steps back). The method repeatability was proved by repeating the experiment hundredfold (Table 1).

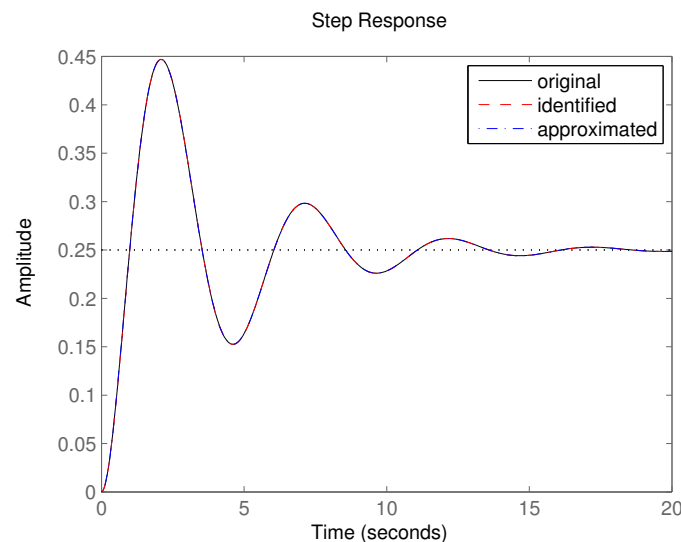


Figure 13. The step response of original, identified and approximated system [49].

Table 1. Exploratory statistics corresponding to hundredfold repeating of simulation [49].

		Y_1	Y_2	Y_3	
original		-2.31816	1.83381	-0.472367	
	mean	-2.31318	1.82481	-0.467787	
identified	median	-2.31390	1.82610	-0.468444	
	std	0.00341	0.00618	0.003142	
	min	-2.31769	1.79844	-0.471932	
	max	-2.29859	1.83297	-0.454380	
		U_1	U_2	U_3	Fitness Function Value
original		0.0264792	0.00182614	-0.0174828	
	mean	0.0262743	0.00210919	-0.0174222	1.20482×10^{-10}
identified	median	0.0263552	0.00203472	-0.0174471	6.02287×10^{-11}
	std	0.0002218	0.00023706	0.0001634	1.80597×10^{-10}
	min	0.0252438	0.00182211	-0.0180518	1.07770×10^{-12}
	max	0.0266417	0.00311891	-0.0168724	1.24241×10^{-09}

The proposed method was robust enough and repeatable, as it was proved. The next idea was to shorten the computing time using only the fragments of the signal, which is also especially useful,

or to parallelize the identification procedure. Based on our findings, the identification from only the sample of the time response also worked properly, and it was faster [43].

Genetic Algorithms Parameters

The use of the genetic algorithms depended on the GA parameters used and the computing power of the equipment employed. All the experiments were performed with the same computing equipment and the same GA setting.

The GA parameters used were set as follows

- Crossover probability—0.9,
- Mutation probability—0.2,
- Population size—31,
- Offspring size—30,
- Selection—rank weighting,
- Crossover—interpolation between two individuals [50],
- Mutation—movement on each dimension by $\pm 0.1 \times$ random number.

A combination of the resulting value of fitness function 1×10^{-12} and 100,000 generations was chosen as the stop criteria for the demonstration of the proposed identification method accuracy. The stop criteria were used independently; if one of them set in, the identification process was stopped. These stop criteria were set to achieve remarkably high accuracy, which could be considered as unnecessary. If it was necessary, according to the accuracy required, the stop criterion could be set in another manner, and the computational time was going to be significantly shorter. For the second-order system, the real computation times were approximately around 1 s for 60,000,000 iterations of the GA conducted on a regular computer with a 2 GHz processor. The convergence speed is shown in Figure 14.

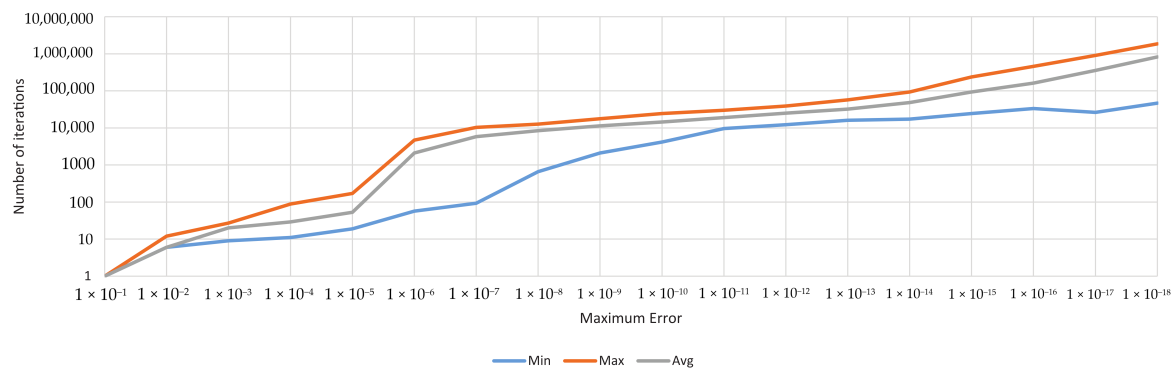


Figure 14. The convergence speed—the accuracy in the point of view of necessary iterations [42].

The only criterion for the use of a system of a high order was the computing time, which could be improved using more powerful hardware. The future work in this area is going to focus on the parallelization of the problem. The next question is whether genetic algorithms could be replaced by another method from the artificial intelligence area, such as a bee colony, ant colony algorithms, or particle swarm optimization. All these propositions were viable. It was decided to use genetic algorithms because they were assessed as the best known in the classic industrial area, where doubts about artificial intelligence approaches persisted.

3.3. Controlling System Design—ES2

The last separate module to be simulated was the module for the PID controller parameters design. The simulation of ES2 was performed using a system with the transfer function:

$$G_{ES2}(s) = \frac{1}{3s^2 + 12s + 15}. \quad (25)$$

According to the design of ES2 presented in the paper (see Section 2.3.1), it can be stated that parameter a_2 was neither XS nor Small, a_1 was more Small than Medium and a_0 was Medium rather than Large. ES2 was designed in three versions compared with the traditional PID controller's corresponding conventional design method (Figure 15)—the Ziegler–Nichols step response method (Figure 15–upper), a combination of Ziegler–Nichols methods (Figure 15–middle) and the Chien, Hrones, and Reswick setpoint response design method with a 0% overshoot in a variant with a 0% overshoot (Figure 15–lower). Comparison of the time response for the unit step of all versions was conducted.

The difference in the overshoot of the time response of a closed feedback control loop for the unit step (Figure 15) was assessed as 24.09%, or 24.375 p.p. and 11.378 p.p. when comparing classic and expert design methods (Table 2) of conventional PID controllers on behalf of expert methods. The difference observed in settling times is shown in Table 2. The settling times in the first two versions were noticeably shorter. In the last option (CHR design method), the settling time was longer, but, with the benefit of the time response, almost without an overshoot.

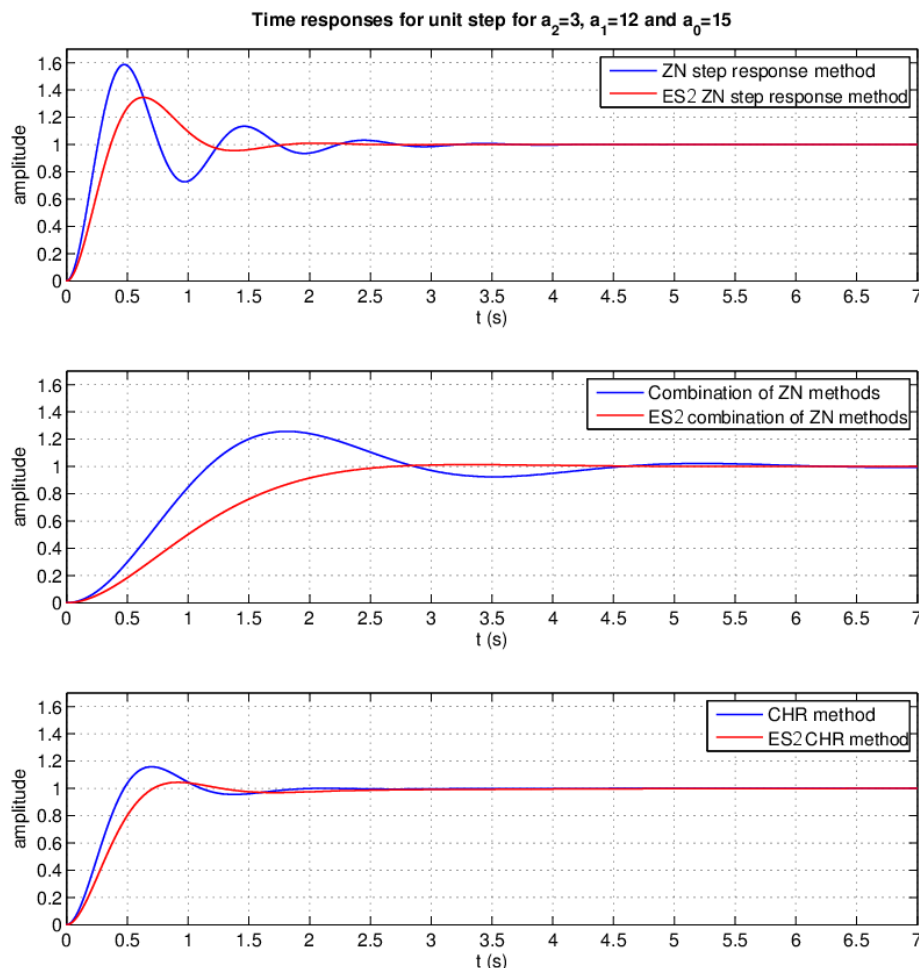


Figure 15. Time responses comparison of the closed feedback loop with a PID controller with parameters determined using the created ES2 and classic appropriate methods for a unit step for $a_2 = 3$, $a_1 = 12$ and $a_0 = 15$ [49].

Table 2. Overshoot and settling time in percent of step response of controlled system with parameters $a_2 = 3$, $a_1 = 12$ and $a_0 = 15$ [49].

Design Method		Overshoot (%)	Settling Time (2% Standard) (s)
ZN step response method	classic	58.69	2.5934
	expert	34.60	1.6355
Combination of ZN methods	classic	25.72	5.4623
	expert	1.345	2.4214
CHR method	classic	15.86	1.6975
	expert	4.482	2.1660

3.4. Intelligent Controller

A general design of the intelligent controller with the artificial intelligence approach is proposed. The intelligent controller concept contained three main parts: quality monitoring, an identification system, and a PID controller design system. The work aimed to show the use and efficiency of artificial intelligence methods in controlling the issue. After verification of the independent modules, all parts (ES1, IS and ES2) were connected, creating an intelligent controlling system according to the schema in Figure 1. The verification focused on demonstrating the efficiency of the proposed intelligent controller.

The re-adaptation strategy using ES1 was briefly described above. For the sake of clarity, the re-adaptation process is shown in Figure 16. The identification procedure (using IS) was continuous (the data were obtained from the sensors continuously, in predefined time samples), so it meant that, in every moment, it should be able to determine the parameters of the controlled system without a significant delay. The ongoing checking of the difference of the required and controlled value (the relative overshoot) of the timing was performed and stored in the memory. In addition, the settling time was measured and stored in the memory after every quick change in the required (step change) or controlled value. After obtaining both values (inputs of ES1), the score was assessed. According to the value of the score,

$$Score \left\{ \begin{array}{ll} \geq 2 & \text{do not re-adapt} \\ < 2 & \text{re-adapt} \end{array} \right\}, \quad (26)$$

re-adaptation of the PID controller was conducted due to the parameters identified by the IS system. ES2 acted based on the decision made by ES1 on whether the re-adaptation was needed and worked with the knowledge obtained from the identification system about the parameters of the controlled system.

The simulation was performed using timing (Figure 17), where all critical moments and information were written. ES2 was used in one of its three versions—the version with the Ziegler–Nichols step response method.

The simulations began with the controlled system (S1) with transfer function

$$G_{S1}(s) = \frac{1}{0.5s^2 + 17s + 22}, \quad (27)$$

and its step response in time t_A . The parameters of the corresponding controller with transfer function

$$G_{R1}(s) = 149 \left(1 + \frac{1}{0.37s} + 0.09s \right), \quad (28)$$

were designed using ES2. In the timing, the overshoot of 14.7% was observed and settling time t_{sf1} of 1.4 sec was obtained from measurement. The corresponding score was evaluated using ES1 as

$$score_1 = 2.01 > 2. \quad (29)$$

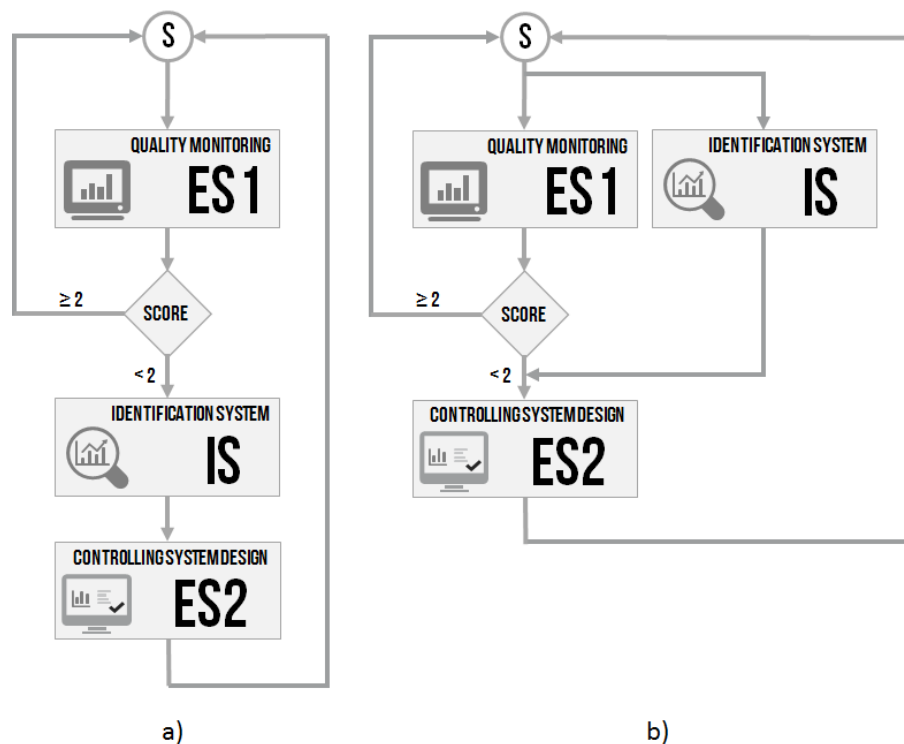


Figure 16. Flowchart of the re-adaptation process in two variants (a) with the identification procedure of the parameters of the controlled system after the score evaluation, (b) with the continuous system identification.

The value of $score_1$ was bigger than 2, so it means that the timing was assessed as good enough. Then, at time t_C , the change of the controlled system was performed to

$$G_{S2}(s) = \frac{1}{s^2 + 2s + 4.5'} \quad (30)$$

in the timing, the overshoot equalled 434.6%, which can be seen in settling time $t_{st2} = 1.2$ sec. The quality of process controlling was evaluated as inappropriate using score:

$$score_2 = 1.11 < 2, \quad (31)$$

as inappropriate. The controlled system was identified using IS with the transfer function:

$$G_{S2ident}(s) = \frac{1}{s^2 + 2.00122s + 4.50034'} \quad (32)$$

in 1.232 s (using sampling period $T = 0.1$) with fitness function $J_{S2} = 6.13771 \times 10^{-10}$. The identification had been stopped before the step change in the required value was performed, but if the identification had not been stopped earlier than the step change was made, the identification would have been stopped by this change in the required value.

The re-adaptation of the parameters of the controller was planned in the next change in the required value, which was introduced in $t_F = 6$ s. The re-adapted controller had the transfer function in the following form:

$$G_{R2}(s) = 6.1 \left(1 + \frac{1}{0.58s} + 0.14s \right), \quad (33)$$

where the parameters were again designed using ES2 in the version with the Ziegler–Nichols method. The control process was performed with an overshoot equalling 13.7% and settling time $t_{st3} = 1.2$ s. The appropriate score calculated by ES1 was:

$$score_3 = 2.04 > 2, \quad (34)$$

which corresponded to the satisfactory control timing.

Afterwards, at $t_G = 10$ sec, the sudden change in controlled system G_{S2} caused a return to the system which had the same transfer function as G_{S1} :

$$G_{S3}(s) = \frac{1}{0.5s^2 + 17s + 22}, \quad (35)$$

which was accompanied by the 3.4%-overshoot and settling time $t_{st4} = 2.8$ s and

$$score_4 = 2.00, \quad (36)$$

which, according to the proposed procedure (see Figure 16), was assessed as appropriate.

At time t_I , the step change in the required value was made again; the score using ES1 was inferred as:

$$score_5 = 2.55 > 2, \quad (37)$$

with the inputs of ES1 settling time $t_{st5} = 3$ s and the overshoot = 4.1%. Since $score_5$ was bigger than 2, re-adaptation of the PID controller was not necessary, it was adequately compensated by the original controller $G_{R2}(s)$.

The entire system incorporating all the parts presented worked properly. In the paper, the design is introduced in a general manner so that it can be modified to another family of a controlled system according to the expert's knowledge.

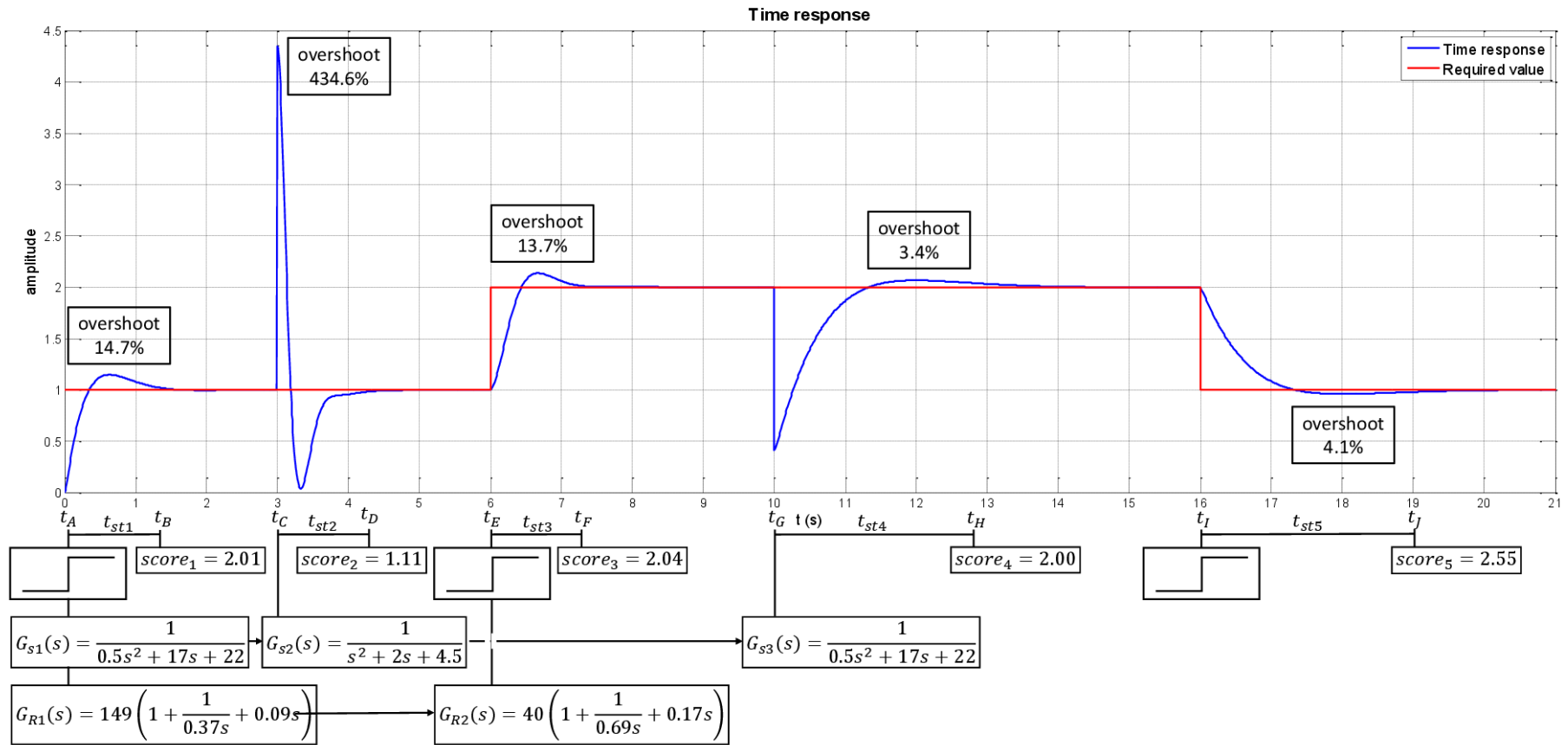


Figure 17. Time course of the simulation of the intelligent controller [49].

4. Discussion

The paper focuses on the application of a modern soft-computing technology. Conventional mathematical methods and unconventional methods of artificial intelligence are integrated into the design of an intelligent controller system. The solution uses information from sensors performing continuous measurements of indicators of the control process. When dealing with the design of the intelligent controller, several goals were pursued, listed in the introduction of the paper.

Its modularity secures versatility and flexibility of the solution. The sub-subsystem for continuous monitoring of the quality of the control process, the subsystem for ongoing identification of the transient function of the controlled system and the subsystem for continuous adaptation of the PID controller are also usable in other applications.

The process monitoring subsystem module and the controller parameter adaptation subsystem module are fuzzy-logic expert systems; their language models are open for extension by other expert rules. Language models are easily understandable to the users.

Language oriented expert modules use Mamdani and Takagi–Sugeno inference mechanisms. Their algorithms are relatively simple. Similar programs are not computationally intensive and contribute to minimizing the required computing power. For this reason, a simple optimization genetic algorithm is also used in the module of continuous identification of the controlled system. Computing modules, as well as an intelligent global controller, can thus be part of embedded control systems. The simplicity of the solution also supports its use in the school teaching process.

The original solution of Mamdani and Takagi–Sugeno fuzzy modules is the choice of structures of their language models, determination of structures and parameters of fuzzy values of their language variables and expert formulation of language rules by their knowledge base. The use of genetic optimization algorithms in the module of continuous system identification is also original.

When designing the system, the control of higher-order systems was considered. The procedures of individual modules and their integration in the intelligent controller system were programmatically implemented in the Matlab and Simulink system. The correctness of the proposed solutions was simulated by testing both modules and their integration in the intelligent controller system.

The effectiveness of artificial intelligence tools was demonstrated by comparing the results of re-adaptation of PID controller parameters. Three conventional adaptation methods were tested, namely the simple Ziegler–Nichols method, the combined Ziegler–Nichols method and the Chien–Hrones–Reswick method. Their mathematical and fuzzy-language models programmatically implemented all three methods. In all three cases, the application of language models was more successful.

In the further development of the presented intelligent controller system, attention will be paid to its implementation in a microprocessor embedded systems.

Author Contributions: Conceptualization, J.N. and M.P.; methodology, J.N. and M.P.; software, J.N.; validation, J.N.; formal analysis, M.P.; investigation, J.N.; resources, J.N.; data curation, J.N.; writing—original draft preparation, J.N. and M.P.; writing—review and editing, M.P.; visualization, J.N.; supervision, M.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Technology Agency of the Czech Republic under the grant no. TN01000024, by the European Regional Development Fund in the Research Centre of Advanced Mechatronic Systems project CZ.02.1.01/0.0/0.0/16_019/0000867 within the Operational Programme Research, Development and Education and by the project SP2020/108 of the Student Grant System, VŠB-Technical University of Ostrava.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

A	Identificator for value Appropriate
CHR	Chein, Hrones and Reswick
COA	Center of Area
e(t)	Control error
ES	Expert System
ES1	Monitoring Expert System
ES2	Controlling System Design
F	Identificator for value Faster
G	Transfer Function
GA	Genetic Algorithms
H	Identificator for value High
IS	Identification System
J	Fitness Function
K	Gain of Proporcional Component
L	Identificator for value Low and Large
M	Identificator for value Medium
MO	Maximal Overshoot
O	Identificator for value Optimal
PID	Proportional–Integral–Derivative
RO	Relative Overshoot
RST	Relative Settling Time
s	Laplace Operator
S	Identificator for value Slower and Small
sec	Time unit second—for clear usage it is not used s, which is used for Laplace Operator
ST	Settling Time
T	Sampling Period
T_d	Time Constant of Derivator
TDKNOW	Time Constant of Derivator obtained by Expert System
T_i	Time Constant of Integrator
TIKNOW	Time Constant of Integrator obtained by Expert System
u(t)	Control variable
U	Laplace Image of Control Variable (System Input)
w(t)	Setpoint—required variable
XL	Identificator for value XLarge
XS	Identificator for value XSmall
y(t)	Controlled variable
Y	Laplace Image of Process Variable (System Output)
ZN	Ziegler–Nichols

References

1. Åström Karl, J.; Hägglund, T. *PID Controllers: Theory, Design and Tuning*, 2nd ed.; Instrument Society of America: Research Triangle, NC, USA, 1995.
2. Ärzén, K.E. An Architecture for Expert System Based Feedback Control. *Automatica* **1989**, *25*, 813–827. [[CrossRef](#)]
3. Åström Karl, J.; Anton, J.J. Expert Control. *IFAC Proc. Vol.* **1984**, *17*, 2579–2584. [[CrossRef](#)]
4. Åström Karl, J.; Anton, J.J.; Ärzén, K.E. Expert Control. *Automatica* **1986**, *22*, 277–286. [[CrossRef](#)]
5. Moore, R.L.; Hawkinson, L.B.; Knickerbocker, C.G.; Churchman, L.M. Expert systems applications in industry. In Proceedings of the ISA International Conference, Houston, TX, USA, 22 October 1984; pp. 22–25.
6. Shirley, R.S. Some lessons learned using expert systems for process control. *IEEE Control Syst. Mag.* **1987**, *7*, 11–15. [[CrossRef](#)]
7. Ünal, M.; Ak, A.; Topuz, V.; Erdal, H. *Optimization of PID Controllers Using ant Colony and Genetic Algorithms*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2012.
8. Kofinas, P.; Dounis, A.I.; Papadakis, G.; Assimakopoulos, M. An Intelligent MPPT controller based on direct neural control for partially shaded PV system. *Energy Build.* **2015**, *90*, 51–64. [[CrossRef](#)]

9. Thangaraju, I.; Muruganandam, M.; Madheswaran, M. Performance Analysis and Experimental Verification of Buck Converter fed DC Series Motor using Hybrid Intelligent Controller with Stability Analysis and Parameter Variations. *J. Electr. Eng. Technol.* **2015**, *10*, 518–528. [[CrossRef](#)]
10. Al-saedi, M.I.; Wu, H.; Handroos, H. Intelligent controller of a flexible hybrid robot machine for ITER assembly and maintenance. *Fusion Eng. Des.* **2014**, *89*, 1795–1803. [[CrossRef](#)]
11. Mishra, P.; Kumar, V.; Rana, K. A novel intelligent controller for combating stiction in pneumatic control valves. *Control Eng. Pract.* **2014**, *33*, 94–104. [[CrossRef](#)]
12. Liu, B.; Ding, Y.; Gao, N.; Zhang, X. A bio-system inspired nonlinear intelligent controller with application to bio-reactor system. *Neurocomputing* **2015**, *168*, 1065–1075. [[CrossRef](#)]
13. Zhou, L.; Chen, G. Intelligent vibration control for high-speed spinning beam based on fuzzy self-tuning PID controller. *Shock Vib.* **2015**, *2015*, 1–8. [[CrossRef](#)]
14. Azali, S.; Sheikhan, M. Intelligent control of photovoltaic system using BPSO-GSA-optimized neural network and fuzzy-based PID for maximum power point tracking. *Appl. Intell.* **2016**, *44*, 88–110. [[CrossRef](#)]
15. Babu, V.S.; Kumar, U.A.; Priyadharshini, R.; Premkumar, K.; Nithin, S. An intelligent controller for smart home. In Proceedings of the 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, India, 21–24 September 2016; pp. 2654–2657.
16. Rahmanian, E.; Akbari, H.; Sheisi, G.H. Maximum power point tracking in grid connected wind plant by using intelligent controller and switched reluctance generator. *IEEE Trans. Sustain. Energy* **2017**, *8*, 1313–1320. [[CrossRef](#)]
17. Jafari, M.; Xu, H.; Carrillo, L.R.G. Brain emotional learning-based intelligent controller for flocking of multi-agent systems. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 1996–2001.
18. Bigdeli, N.; Ziazi, H.A. Design of fractional robust adaptive intelligent controller for uncertain fractional-order chaotic systems based on active control technique. *Nonlinear Dyn.* **2017**, *87*, 1703–1719. [[CrossRef](#)]
19. Han, H.G.; Wu, X.L.; Liu, Z.; Qiao, J.F. Design of self-organizing intelligent controller using fuzzy neural network. *IEEE Trans. Fuzzy Syst.* **2017**, *26*, 3097–3111. [[CrossRef](#)]
20. Golshannavaz, S.; Khezri, R.; Esmaeeli, M.; Siano, P. A two-stage robust-intelligent controller design for efficient LFC based on Kharitonov theorem and fuzzy logic. *J. Ambient Intell. Humaniz. Comput.* **2018**, *9*, 1445–1454. [[CrossRef](#)]
21. Adhvaryu, A.D.; Adarsh, S.; Ramchandran, K. Design of fuzzy based intelligent controller for autonomous mobile robot navigation. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; pp. 841–846.
22. Saxena, A.; Kumar, J.; Deolia, V.K. Design a Robust Intelligent Controller for Rigid Robotic Manipulator System having Two Links and Payloads. In Proceedings of the 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), Mathura, India, 28–29 February 2020; pp. 159–163.
23. Chiu, C.H.; Wu, C.Y. Bicycle Robot Balance Control Based on a Robust Intelligent Controller. *IEEE Access* **2020**, *8*, 84837–84849. [[CrossRef](#)]
24. Heij, C.; Ran, A.C.; van Schagen, F. *Introduction to Mathematical Systems Theory: Linear Systems, Identification and Control*, 1st ed.; Birkhäuser Verlag: Basel, Switzerland, 2007.
25. Ljung, L. *System Identification: Theory for the User*, 2nd ed.; Prentice-Hall: Englewood Cliffs, NJ, USA, 1999.
26. Coley, D.A. *An Introduction to Genetic Algorithms for Scientists and Engineers*, 1st ed.; World Scientific: Hackensack, NJ, USA, 1999.
27. Abonyi, J.; Feil, B. *Cluster Analysis for Data Mining and System Identification*, 1st ed.; Birkhäuser Verlag AG: Berlin, Germany, 2007.
28. Chu, S.R.; Shoureshi, R.; Tenorio, M. Neural networks for system identification. *IEEE Control Syst. Mag.* **1990**, *10*, 31–35. [[CrossRef](#)]
29. Nelles, O. *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2001.
30. Nowaková, J.; Pokorný, M. Double Expert System for Monitoring and Re-adaptation of PID Controllers. In *Innovations in Bio-inspired Computing and Applications*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 85–93.

31. Nowaková, J.; Pokorný, M.; Pieš, M. Conventional controller design based on Takagi–Sugeno fuzzy models. *J. Appl. Log.* **2015**, *13*, 148–155. [[CrossRef](#)]
32. Yazdanbakhsh, O.; Dick, S. A systematic review of complex fuzzy sets and logic. *Fuzzy Sets Syst.* **2018**, *338*, 1–22. [[CrossRef](#)]
33. Jager, R. *Fuzzy Logic in Control*, 1st ed.; Kluwer Academic Publishers: Delft, The Netherlands, 1995.
34. Doerr, B.; Le, H.P.; Makhmara, R.; Nguyen, T.D. Fast genetic algorithms. In Proceedings of the Genetic and Evolutionary Computation Conference, Berlin, Germany, 15–19 July 2017; pp. 777–784.
35. Hu, B.G.; Mann, G.K. A Systematic Study of Fuzzy PID Controllers - Function-Based Evaluation Approach. *IEEE Trans. Fuzzy Syst.* **2001**, *9*, 699–712.
36. Farsi, M.; Karam, K.; Abdalla, H. Intelligent multi-controller assessment using fuzzy logic. *Fuzzy Sets Syst.* **1996**, *79*, 25–41. [[CrossRef](#)]
37. Nilsson, N.J. *Principles of Artificial Intelligence*, 2nd ed.; Morgan Kaufmann: Burlington, MA, USA, 2014.
38. Zou, L.; Wen, X.; Wang, Y. Linguistic truth-valued intuitionistic fuzzy reasoning with applications in human factors engineering. *Inf. Sci.* **2016**, *327*, 201–216. [[CrossRef](#)]
39. Yager, R.R.; Zadeh, L.A. *An Introduction to Fuzzy Logic Applications in Intelligent Systems*, 2nd ed.; Springer Science & Business Media: Berlin, Germany, 2012.
40. Cornelis, C.; De Cock, M.; Kerre, E.E. The generalized modus ponens in a fuzzy set theoretical framework. In *Fuzzy If-Then Rules in Computational Intelligence*; Springer: Boston, MA, USA, 2000; pp. 37–59.
41. Pintelon, R.; Schoukens, J. *System Identification: A Frequency Domain Approach*, 2nd ed.; John Wiley & Sons: New York, NY, USA, 2001.
42. Nowakova, J.; Platos, J.; Snasel, V. Automatic power system identification using genetic algorithms. In Proceedings of the 2014 International Conference on Intelligent Networking and Collaborative Systems, Salerno, Italy, 10–12 September 2014; pp. 133–137.
43. Nowakova, J.; Platos, J.; Hasal, M. System identification acceleration and improvement with genetic programming usage. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–7.
44. Chambers, L.D. *Practical Handbook of Genetic Algorithms: Complex Coding Systems*; CRC Press: Boca Raton, FL, USA, 2019; Volume 3.
45. Miller, F.P.; Vandome, A.F.; McBrewster, J. *Nyquist-Shannon Sampling Theorem*, 1st ed.; Alphascript Publishing: San Carlos, CA, USA, 2010.
46. O'Dwyer, A. *Handbook of PI and PID Controller Tuning Rules*, 3rd ed.; World Scientific: Singapore, 2009.
47. Zhao, Z.Y.; Tomizuka, M. Fuzzy Gain Scheduling of PID Controllers. *IEEE Trans. Syst. Man Cybern.* **1993**, *23*, 1392–1398. [[CrossRef](#)]
48. Guo, Y.; Yang, T. A New Type of Computational Verb Gain-scheduling PID Controller. In Proceedings of the 2010 International Conference on Anti-Counterfeiting Security and Identification in Communication (ASID), Chengdu, China, 18–20 July 2010; pp. 235–238.
49. Nowaková, J. Knowledge-Based Adaptation of Controllers. 2017. Available online: <https://dspace.vsb.cz/handle/10084/122038> (accessed on 9 August 2020).
50. Collette, Y. Crossover GA Default Function. Available online: https://help.scilab.org/docs/6.1.0/en_US/crossover_ga_default.html (accessed on 15 July 2020).

