International Journal of Modern Manufacturing Technologies
ISSN 2067–3604, Vol. XIV, No. 2 / 2022
https://doi.org/ 10.54684/ijmmt.2022.14.2.188

AN AUTOMATIC LIQUID LEVEL CONTROL SYSTEM WITH THE USE OF THE ARDUINO MICROCONTROLLER AND STRAIN GAUGES - ANALYSIS OF APPLICATION POSSIBILITY IN MANUFACTURING PROCESSES

Marek Płaczek[1], Maciej Kaźmierczak[2]

[1]Silesian University of Technology, Faculty of Mechanical Engineering, Department of Engineering Processes Automation, and Integrated Manufacturing Systems, Konarskiego 18A, 44-100 Gliwice, Poland
[2] NGK Ceramics Polska Sp. z o.o., J. Gutenberga 6, 44-109 Gliwice, Poland

Corresponding author: Marek Płaczek, marek.placzek@polsl.pl

Abstract: The aim of the work was to present the possibility of automatic regulation of the liquid level in the tank with the use of the Arduino platform, strain gauge measurements and pump operation regulation by means of a PID controller. The possibility to program the Arduino and implementing the algorithm of the PID controller was presented. To be able to use the PID controller, the controller was tuned using the II Ziegler-Nichols method and the manual method. To check how the applied controller works in automatic liquid level control and how its settings affect the operation of the system, tests were carried out comparing the operation of various configurations of the PID controller in various operating conditions. The work carried out showed the possibility of developing a low-cost automatic liquid level control system, which can be successfully used in manufacturing processes.
Key words: *Strain gauge, automatic control, measuring device, microcontroller, Arduino, PID controller.*

1. INTRODUCTION

Automatic control systems are widely used in industry, but also in domestic applications. They are usually built based on ready-made solutions prepared by manufacturers. This makes such solutions expensive and has limited modifiability. Such systems must be carefully designed, equipped with sensor systems and control algorithms to ensure their safe use [5, 6]. A critical issue is also the verification of the possibility of implementing automatic regulation of systems using low-budget solutions while ensuring stable operation of the systems and proper quality of regulation. This issue is addressed in this study.
Diverse types of strain gauges can be used in such systems. They allow to measure, for example, pressure force, weight, or pressure in tanks. Therefore, they are widely used in industry to measure the amount of raw material or liquid in a tank. Such sensors placed under the tank measure the deformation of the beams and on this basis the weight can be read by means of appropriate transducers and software usually implemented in a PLC controller. In an analogous way, strain gauges are used to measure deformations in several types of metal structures. This is used, for example, in rail transport, where such sensors are used to measure the deformation of bridges during the passage of trains, which makes it possible to determine the strength of the bridge and to measure deformation in pipes and industrial machines. In automatic systems for reading from sensors and for information processing and control, industrial solutions are usually used, i.e., PLC controllers, which allow you to easily connect sensors, implement a program with a ready PID algorithm and control actuators. However, you can also use fewer industrial solutions, using ready-made sets with a microcontroller. Such a solution requires writing your own algorithm for the PID controller.
The aim of the work was to create a laboratory stand presenting the possibilities of automatic regulation of the liquid level in the tank with the use of the Arduino platform, strain gauge measurements and control of pump operation using the PID controller.
The concept of strain gauges refers to the measurement of stresses, but nowadays it is understood as measurements using metal resistance strain gauges. Strain gauge otherwise is an electrical method of measuring non-electrical quantities. Strain gauge measurements are usually carried out using metal strain gauges. Metal strain gauges are made by etching a metal foil on an insulating substrate and then applying it to the object that is the purpose of the measurements. For measurements using electro resistance strain gauges, systems consisting of an input circuit, an amplifier and an output circuit are usually used. The deformations of the objects are small, so the resistance changes are small.
Appropriately sensitive instruments should be used to record these changes. The input circuit is usually a Wheatstone bridge of strain gauges and usually allows

initial adjustments and sensitivity balancing. Amplifiers are used to amplify and process the signal. The output system consists of recording and indicator devices. Strain gauge measurements are widely described in numerous bibliographic items and find practical application in a lot of industries and technical means.

The article [4] presents research methods for determining the compression curves of structural materials during impact loads using the Hopkinson rod. One of the topics of the work was to present a laboratory stand for evaluating the mechanical characteristics of materials at high strain rates. This station in its construction includes electro-resistance strain gauges. These strain gauges were used to measure the course of elastic waves that were generated because of the impact of the projectile. In the further part of the work, the author discussed the possibility of miniaturization of the stand, which allows to achieve higher strain rates, which is required for some applications. The apparatus at the Motor Transport Institute was also presented, which was used to determine the characteristics of stress as a function of deformation of selected alloys. The obtained data made it possible to assess the sensitivity of the tested materials to the strain rate.

Another interesting application of strain gauges is their use in car seats [9]. The strain gauge can detect whether a person is sitting in the passenger seat. There are several methods to realize passenger detection. The described method involves the use of strain gauges attached to the chair structure. The purpose of occupant presence detection is to decide whether to deploy the airbags in the event of an accident, however, simply detecting the presence of any weight on the seat may not be sufficient as there may be an object on the seat or a child in the child seat. Actuation of the airbag in such a case may be unnecessary and even harmful. For this reason, the authors of the patent used strain gauges that not only make it possible to determine whether there is any object on the passenger seat, but also to determine the size of the object. One of the options for mounting the sensors is to place them in the seat cushion, but this method does not measure the pressure on the seat back, which makes the determination of weight imprecise. Another possibility is to place the sensors between the seat and the frame, in the places of attachment. This allows weight to be measured on the backrest but requires a redesign of the chair. The patented solution proposes placing strain gauges in a classic car seat structure. They are rigidly attached to part of the supports to consider the effect of pressure not only on the seat surfaces but also on the backrest, which allows for accurate weight measurement.

In this paper, it is proposed to combine strain gauge measurements with the system of automatic regulation of the liquid level in the tank with the use of a PID controller, whose operating algorithm was implemented in the Arduino microcontroller. The use of PID controllers is very wide. Starting from position control in drives, ending with complex systems, considering many variables.

An example of using the PID controller is presented in [3]. The authors describe the use of the PID controller and the modified PI controller in the control of an induction furnace. The research was conducted on a computer model created in the MATLAB - Simulink program. For the tests, the transfer function model of the object was determined based on the dynamic characteristics, and then a simulation model of the control system was created. This model was then implemented in two versions in the MATLAB - Simulink environment with control using a classic PID controller, which was used as a reference to compare it with a modified, predictive PI controller. In the next stage, the quality of regulation during simulation tests was analysed. Two integral indices were adopted as the assessment criteria - control signal dynamics and control quality. Simulation tests were conducted in two variants, assuming ideal conditions and in the presence of a sinusoidal and rectangular interference signal. The disturbance frequency was 0.002 Hz, and the amplitude was from 1% to 5% of the set signal value. The study of the analysed control model shows that in the case of ideal conditions, both controls performed similarly - the quality indicators had similar values. On the other hand, in the case of disturbances, it can be concluded from the indicators that a classic PID controller provides better control.

PID controller application in pneumatic drive systems is presented in [10]. Paper presents the process of selecting parameters in the process of adjusting the position of a rod less actuator. Author’s goal was to conduct research to select the PID controller settings, under the assumptions of shortest possible reaction time of the actuator to changes in position setting, positioning accuracy at the level of 2mm, system stability and presentation of the influence of control settings on the system behaviour. Influence of other parameters on stability and safety of pneumatic systems is also presented in [7].

It is also possible to implement a PID controller in the Arduino microcontroller. Such a solution was presented in [1]. The authors created a system consisting of LabView and Arduino, which is used to monitor and control the temperature of the heating element using a thermocouple. In this project, Arduino was used as a controller to read values from a thermocouple and as a control device. This is due to the ease and simplicity provided by Arduino, based on the ATmega328 microcontroller. It has analogue inputs and digital inputs and outputs, including PWM outputs, which allowed for easy use of this board in the project. A thermocouple was used to read the temperature, while the possibility of using a thermistor or an RTD (Resistance Temperature Detector) sensor was also investigated. The PID algorithm has been implemented in the LabView environment and a graphic page of the program has been created, which allows you to track, among others, temperature values from the sensor and set the regulator parameters. The LINX toolkit was used for communication between the Arduino and the LabView environment running on the computer. The work proves that it is possible to create a temperature control system or other similar systems using simple and cheap controllers, such as Arduino, by connecting them with virtual instruments.

In this work, an attempt was made to implement a similar problem and to analyse the possibility of using a solution based on the regulation of the liquid level in the tank with the use of strain gauges and the PID controller implemented on the Arduino microcontroller.

2. RESEARCH METHODOLOGY

Proportional-integral-derivative (PID) controllers and their various configurations are the most popular method of control in automatic systems. This is due to the ease of their implementation and a simple principle of operation. This technology has been gaining popularity since the 1940s, when automation began to be used increasingly. Currently, when talking about PID controllers, we talk about their applications based on microprocessors. Earlier, however, these were analogue systems, and even mechanical or pneumatic. The basic difficulty when working with PID controllers is the need to tune them, which determines whether such a controller will work correctly and satisfactorily. However, practice shows that regulators are often not finely tuned, which affects the efficiency of automatic systems, [2].

The PID controller consists of three terms, proportional (P), integral (I) and derivative (D). Depending on which term will be used, we distinguish P, PI, PD and PID configurations. The mathematical models of the terms are as follows:

- Proportional regulator P:

$$C_{(s)} = K_p, \tag{1}$$

- Proportional-integral controller PI:

$$C_{(s)} = K_p\left(1 + \frac{1}{sT_I}\right), \tag{2}$$

- Proportional-derivative controller (ideal) PD:

$$C_{(s)} = K_p(1 + sT_D), \tag{3}$$

- Proportional-derivative controller (real) PD:

$$C_{(s)} = K_p\left(\frac{sT_D}{1 + sT_D}\right), \tag{4}$$

- Proportional-integral-derivative (ideal) PID controller:

$$C_{(s)} = K_p\left(1 + \frac{1}{sT_I} + sT_D\right), \tag{5}$$

where:

K_p - amplification factor of the P controller,

T_I - integration time constant (doubling time),

T_D - derivative time constant (lead time).

Each of the elements of the PID controller is characterized by a certain action:

- Proportional part - reduces the error in steady state, can shorten the control time, but increases overshoot (exceeding the set value). A P-type controller will never have zero error in steady state.
- Integral part – responsible for the elimination of the steady-state error, extends the control time and increases the overshoot.
- Derivative part – reduces the regulation time and reduces overshoot [8].

The regulator must be fine-tuned to function properly. Therefore, it is necessary to choose its settings accordingly. There are many methods of setting settings. Two methods were developed by Ziegler and Nichols.

- The second Ziegler-Nichols method is an experimental method and does not require knowledge of the dynamic model of the object. It is performed in a closed system. As a result, this method provides an overshoot of 20% and a short settling time. The steps for this method are as follows:
- in the controller, the integral ($T_I = \infty$) and derivative action ($T_D = 0$) must be turned off,
- increase the k_p gain until constant oscillations occur,
- measure the oscillation period T_{osc} and read the critical gain $k_{r,}$
- set values should be taken in accordance with the table available in the literature [11].

The developed station was designed to automatically regulate the liquid level in the tank. It is presented in Fig. 1. For this purpose, several elements were used that allowed the implementation of this task.

There are two liquid tanks on the stand. One of them was placed at the end of a metal beam. There are foil strain gauges on the beam, which are used to measure its deformations. The other end of the beam was attached to a metal frame. Next to the tank there are two pumps and a second tank. The pumps pump water between the tanks so that the first tank contains a certain amount of liquid. The control of the pumps is carried out using the Arduino Mega board placed on the station.

Strain gauge beam - Wheatstone bridge - is used to measure the weight of the liquid tank. It is built of foil strain gauges and resistors (or the strain gauges themselves). An external power supply is connected to two branches of the bridge, the voltage on the other two branches is measured, which is then converted into a weight value. The foils forming the bridge are glued to the metal beam on which the liquid tank is placed. Depending on the configuration, one foil can be glued to an unloaded beam, which will eliminate disturbances caused by the influence of temperature. The beam bending under the influence of the mass of the liquid causes the foil to bend, which translates into a change in the resistance of the foil, and thus the measured voltage. However, these changes are small, which is why the HX711 strain gauge converter was used - built on the patent of AVIA Semiconductor, it is a 24-bit analogue-to-digital converter. It was designed for scales and industrial systems based on strain gauge beams.

The Arduino MEGA board is the main control element of the regulation process. The input of the board is supplied with voltage from the transducer (voltage - directly or current - with the use of an appropriate transducer), which is converted into the weight value. Then, the PID algorithm implemented on the board properly controls the board's PWM outputs, depending on whether there is too much or too little liquid, it controls the appropriate pump. It is not possible to control the pump motors directly due to the current limitation of the Arduino board outputs, so the L298N driver module had to be used.

LCD display - a two-segment, 16-character alphanumeric display, its task is to display information to the user. It shows a welcome screen, an option selection screen, a screen with the current PID parameters, and a screen with PWM signal values and offset. It is possible to connect the screen directly to the Arduino controller, but it requires the use of many pins, which is why the i2C module was chosen.

I2C converter - allows to operate the LCD screen with the HD44780 controller using two lines - SCL and SDA, which simplifies the way of connecting the screen. It also requires +5V and GND to be connected. In addition, there is a potentiometer on the converter to change the contrast of the display. To control the screen using the converter, it is required to use the free "LiquidCrystal_I2C.h" library.

L298N driver module - DC motor driver, allows you to control two DC motors or one stepper motor. It is powered with the voltage up to 24V and has a maximum output current of 2A. The speed control is done using the PWM signal, in this case coming from the Arduino board, each of the marked "ENA" and "ENB" pins corresponds to one motor. In addition, it is necessary to properly control the pins IN1, IN2 - the first motor and IN3, IN4 - the second motor. High or low state (0 or 5V) is given to the pins. This allows to change the direction of rotation of the motor and slow the motors in two modes - fast and free.

The numeric keypad - a 4x4 keypad - is connected to the eight pins of the Arduino. The keyboard combined with the display allows to switch between screens and change the parameters of the PID controller.

DIN15W12 power supply - power supply with 200-240 VAC input and 12 VDC output. The power supply is mounted on a DIN rail. The maximum output current is 1.25A, which allows to power all electrical components of the entire system.

Overcurrent circuit breaker - circuit breaker with B characteristic and rated current of 6A. The fuse protects the system against a short circuit and allows for quick switching on and off the entire system without having to remove the plug from the socket.

Water pump powered by 12V DC. The brushless DC motor is responsible for the operation of the pump. The pump allows to control the flow rate by adjusting the voltage. The maximum flow is 110 l/h.

Peristaltic pump powered by 12V DC. The construction of such a pump allows for operation in two directions, which is why the pump was used to manually add or pour water from the tank. It is controlled by a three-position switch. The efficiency of the pump is low, but sufficient to introduce disturbances into the system.

In addition to the above-mentioned elements, a three-position switch is used to control the system, which allows to control the peristaltic pump, i.e., turn it on to add or pour water from the tank. A bistable button with actuation backlight was also used. The button is used to start the PID algorithm. This allows to change the regulator settings with the liquid level control switched off.

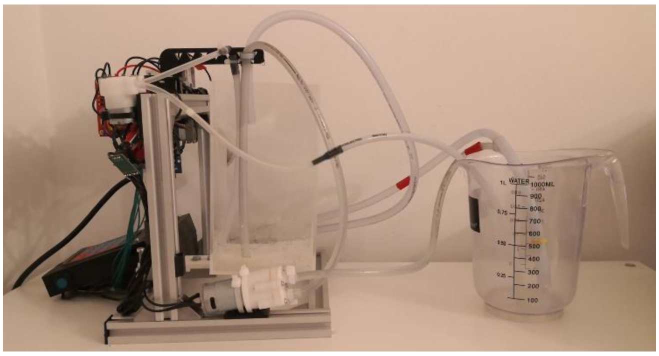


Fig. 1. Automatic liquid level control station

2.1 System control program

The program controlling the operation of the system was written in the Arduino IDE based on the C/C++ language. The program can be divided into several functions:

a) Support for the PID algorithm - the PID algorithm is responsible for controlling the pumps that add or pour liquid from the tank depending on the error, so the difference between the set value and the measured value. For the algorithm to work properly, it is necessary to read the weight from the HX711 converter. The value read is compared with the set value, which is zero by default. The value 0 refers to the system at the time of starting the station. The amount of liquid at start up is the reference value that the system will strive for. Next, calculations are made in accordance with the PID algorithm. To perform them, apart from the error, it is necessary to enter the values of the gains of individual controller elements: K_p, K_i and K_d. By default, settings determined by the experimental method are introduced in the program. The control signal is the sum of three components: proportional component, integral component, and derivative component. The introduced algorithm performs the operation of the ideal algorithm according to the relation (5). The implemented PID controller works discreetly. The sampling and execution of the algorithm depends on the speed of the processor. Therefore, it is necessary to include the sampling time in the algorithm. The code algorithm looks like this:

- proportional term:

$$P = K_p * e, \qquad (6)$$

- Integral controller:

$$I = K_i * e_{sum}, \qquad (7)$$

- Derivative controller:

$$D = K_D * \frac{e - e_{n-1}}{h}, \qquad (8)$$

- Control signal:

$$Y = P + I + D, \qquad (9)$$

where:

$$e_{sum} = e * h + e_{sum},$$

$$K_i = \frac{K_p}{T_i}, \qquad K_D = K_p * T_D, \qquad (10)$$

$h -$ sampling period.

The user of the device can freely change the amplification using the numeric keypad. This also allows you to freely modify the configuration of the regulator. Entering the value 0 as the pre-set setting causes switching off the individual element. Comparison of the operation of different configurations of the regulator is presented in the further part of the work. There are several constraints in the PID algorithm. A limitation of the PWM control signal has been introduced. The signal ranges from -255 to 255, a negative value controls one pump, and a positive value controls the other pump. An anti-windup system has also been added, i.e., limiting the summation of the error when the control signal reaches the minimum or maximum of its capability. During system operation tests, a large delay in pump control was also noted due to the wide range of the PWM signal in which the motors do not rotate. The PWM control signal assumes a value in the range of 0 - 255 for one of the motors and 0 - -255 for the other motor. Which corresponds to a voltage of 0 - 12V, but at low voltage values, corresponding to a range of about 0 - 85 PWM, the motors do not respond. Therefore, this zone has been restricted. The limit of the controller output value and the offset of the control signal have been introduced. For example, in the case of one of the pumps, this was done as follows: the controller output limit is set to 175, while the offset set to 80 is added to the output value. Thus, the controller operates in the range of 0 - 175, and the output in the range of 80 - 255. Such control causes that the control signal from the lowest value has an effective effect on the system. Tests were conducted comparing the operation of the regulator with and without the non-operation zone. The results are presented in the further part of the paper.

b) PWM scaling function - depending on whether the system is to operate with dead band reduction, the offset value is added to the control signal.

c) Motor control function - depending on whether the control signal is positive or negative, the pump adding water or the pump pouring water is turned on. This is done by properly setting the digital outputs of the microcontroller, connected to the module controlling motors. The flow rate is regulated by the PWM signal from the Arduino output connected to the ENA and ENB inputs of the L298N controller.

d) Keyboard operation - The numeric keypad has sixteen keys, numbers 0 - 9, letters A, B, C, D, and special characters * and #. The program performs the reading of individual buttons and based on them, the control of the display and the introduction of new settings of the PID controller parameters. In the program loop, it is checked which button was pressed, it is remembered and, depending on this, the appropriate program steps are performed.

e) I2C display support - a ready-made library was used to support the display, allowing the display to be operated through the I2C converter using only two signal lines. The screen allows you to display two lines of sixteen characters each. Due to this limitation, different information is displayed depending on the key pressed. The screen must be constantly refreshed to display the correct control signal and error values, which change at a high frequency.

f) Graphs function - allows to draw selected values on a graph in Telemetry Viewer V0.7.

3. SELECTION OF REGULATOR SETTINGS AND QUALITY TESTING OF SYSTEM REGULATION

Using the developed system, a study was conducted comparing the tuning of the PID controller using the II Ziegler-Nichols method consisting in determining the critical gain, and on its basis the controller and manual tuning. Both methods are experimental methods and do not require detailed knowledge of the object model. Fig. 2 shows the control in the system at the limit of stability.

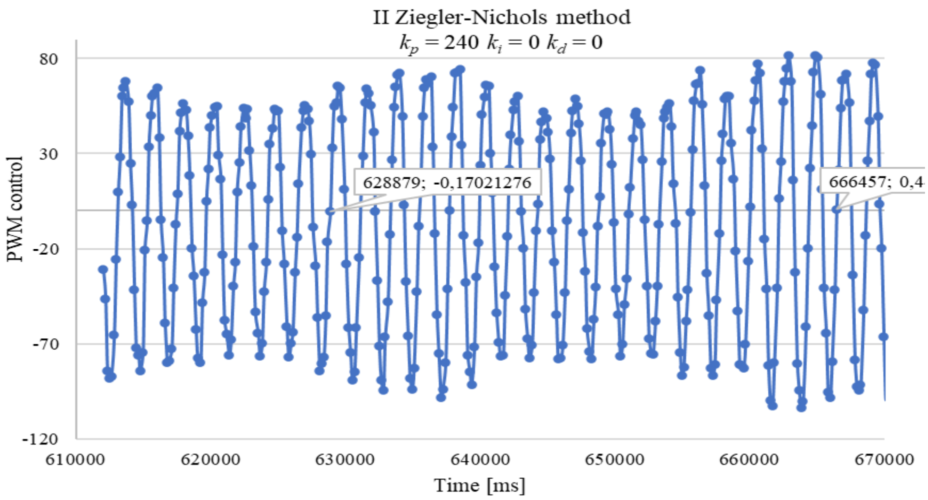


Fig.2. System at the limit of stability

In Fig. 2, two signal samples are marked as close as possible to the intersection with the x axis. To determine the oscillation time, the number of periods and times were read: $T_1 = 628870$ ms, $T_2 = 666457$ ms, $n = 17$ and the oscillation time was calculated:

$$T_{osc} = \frac{T_2 - T_1}{n} = 2.211s. \tag{11}$$

Then, according to [8], the controller settings were determined and are presented in Tab. 1.

Table 1. Regulator settings determined by II Ziegler-Nichols method

	K_p	T_I	T_D
P	0.5 k_r = 120	-	-
PI	0.45 k_r = 108	0.85 T_{osc} = 1.879	-
PID	0.6 k_r = 144	0.5 T_{osc} = 1.1055	0.12T_{osc} = 0.2653

Based on the obtained PID controller settings, the amplification of individual parts of the controller was determined, rounded to integer parts. Table 5 lists the determined controller settings.

Table 2. Regulator gains

	K_P	$K_I = K_P/T_I$	K_D=K_P T_D
P	120	-	-
PI	108	57	-
PID	144	142	38

The settings were verified on the system. The control time, the rise time and the maximum overshoot were determined to evaluate the regulation. Due to the sampling period, it was not possible to read the values at exact points, therefore the determined indicators are a certain approximation. Evaluating the settings consisted in changing the set value Y_{sp} from 0 to 5, this value corresponds to adding about 100 ml of liquid to the tank. This value was not scaled in relation to the liquid weight, as it would require additional calculations and loading the scale with a reference weight.

The maximum overshoot was determined as the difference between the maximum response of the system and the set value:

$$M_{max} = Y_{max} - Y_{sp} \tag{12}$$

The control time t_r was determined as the time after which the system response is within +/- 2% of the set value. The rise time t_n was determined as the time during which the system response changes from 10% to 90% of the set value. The response of the system with P, PI and PID controller settings is shown in Figures 3*a*, 3*b* and 3*c*, respectively.

The determined control quality indices of three regulators with parameters selected by the II Ziegler-Nichols method are presented in Table 3.

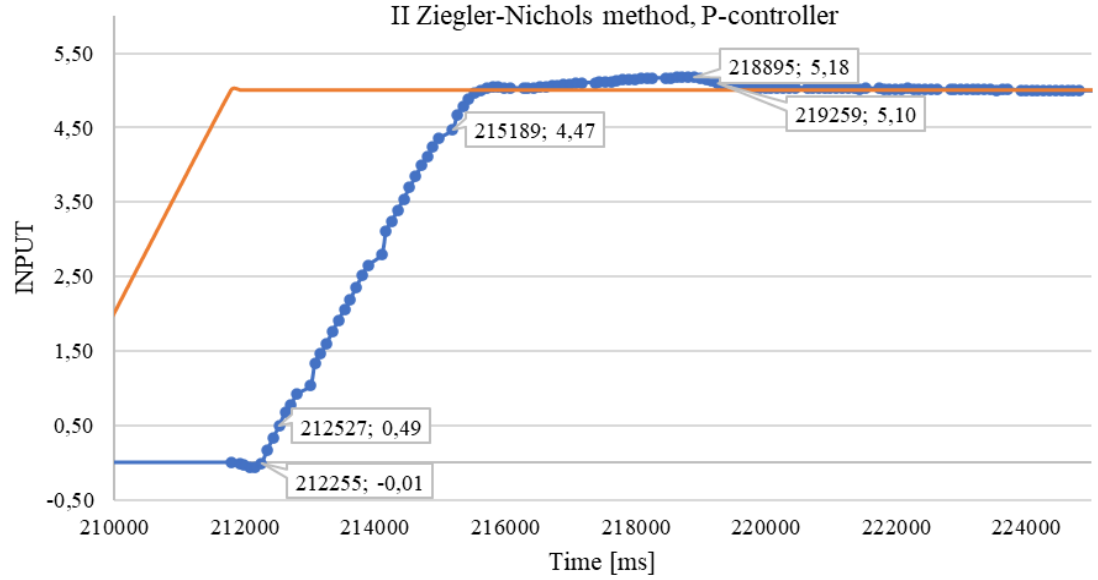
Table 3. Indicators of system regulation quality

	P	PI	PID
M_{max}	0.18	1.41	0.81
t_r [s]	7	7.276	8.458
t_n [s]	2.662	2.571	2.662

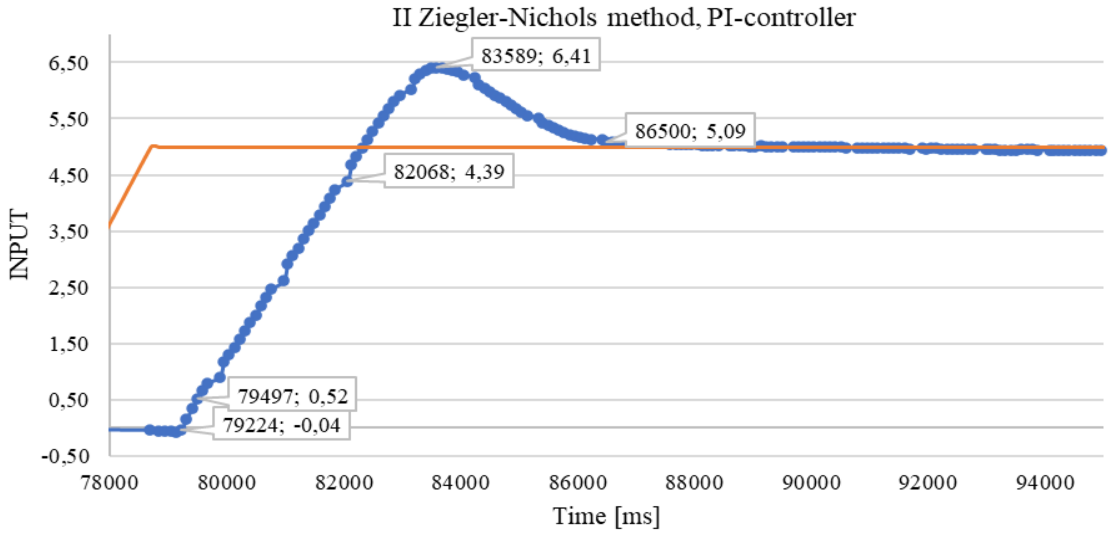
Based on the determined indicators, it can be concluded that particularly good control parameters were obtained using the P-type controller. It had a small overshoot and a short adjustment time. In the case of the PI controller, the greatest overshoot occurred, the control and rise times were like the P controller. The longest control time was recorded in the PID controller. It should be noted that the test was conducted in the way of changing the set point with no disturbances and in this case the P-type controller works very well.

In the next test, the set point was not changed, but a permanent disturbance was introduced, liquid is pumped out of the regulated tank at a constant flow rate. In this case, the system acts as a follow-up system. Fig. 4 shows the response of the system with a P (*a*), PI (*b*) and PID (*c*) controller and constant disturbances, respectively.

a)



b)



c)

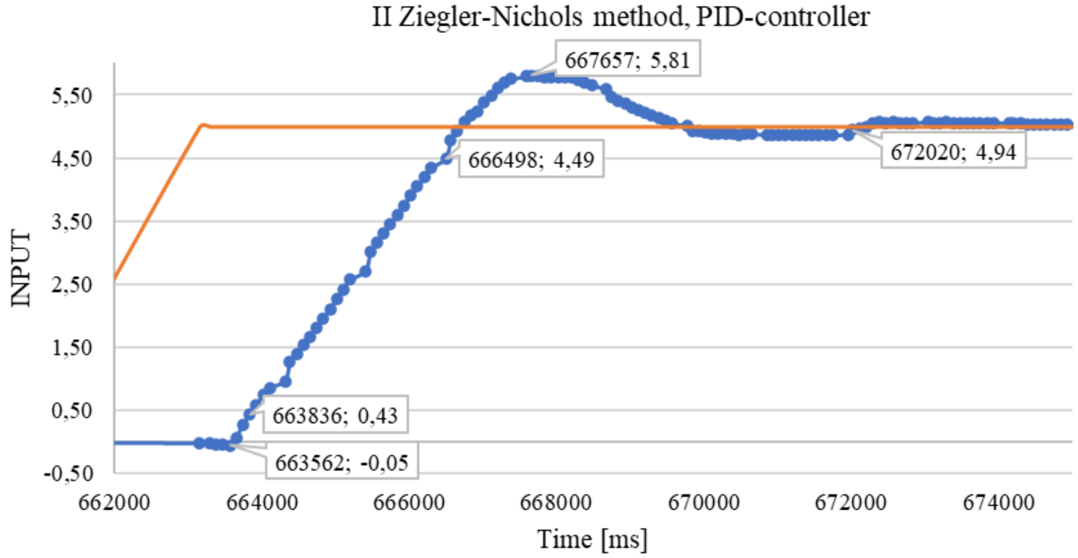


Fig. 32. System response with P (*a*), PI (*b*) and PID (*c*) controllers

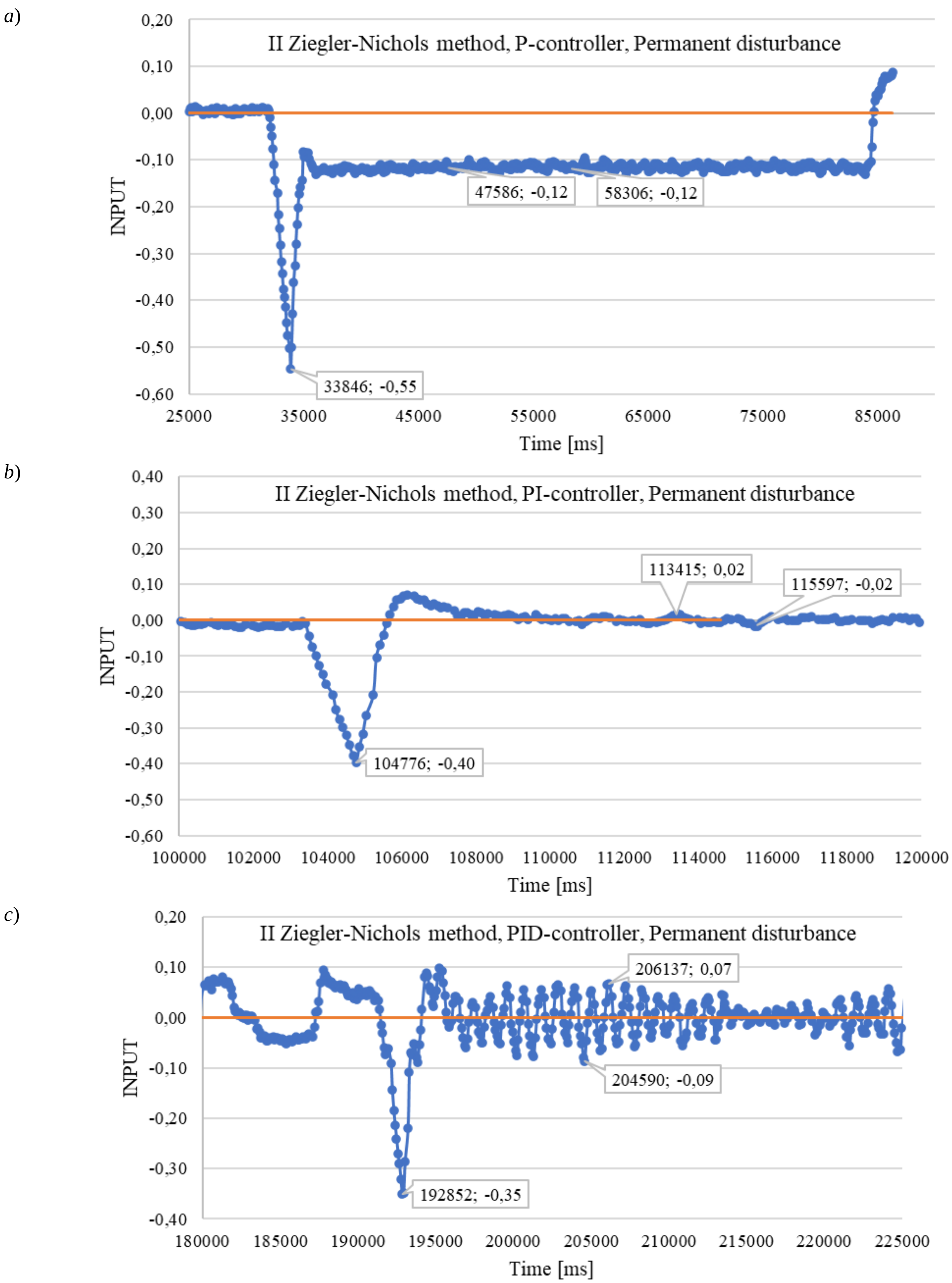


Fig. 3. System response with P (*a*), PI (*b*) and PID (*c*) controllers under constant disturbance

With a constant disturbance, the greatest defect of the P-type controller is visible, i.e., the steady-state error will always be different from zero. In this case, the steady-state error is about -0.12. The maximum deviation from the set level related to the sudden appearance of the disturbance was -0.55.

In the case of the PI controller, the maximum deviation from the set level has been reduced to -0.4, and the steady-state error does not exceed -/+ 0.02.

In the case of the PID controller, a faster reaction of the system can be seen, the maximum deviation at the appearance of the disturbance was -0.35, while in the steady state, oscillations appeared, and the error increased compared to the PI controller. This may be due to the use of an ideal PID controller and the lack of a low-pass filter, because of which even slight changes in measurement cause substantial changes in control.

The next stage of evaluating the system operation was to compare the operation of the system with the dead zone switched off and switched on. For this purpose, the control offset has been removed in the program. Then, once again, the settings were selected using the II Ziegler-Nichols method. PI type controllers were compared. In such a system, the critical amplification was 570. Figure 5 shows the control in the system during the determination of II Ziegler-Nichols method settings without correction of the dead zone.

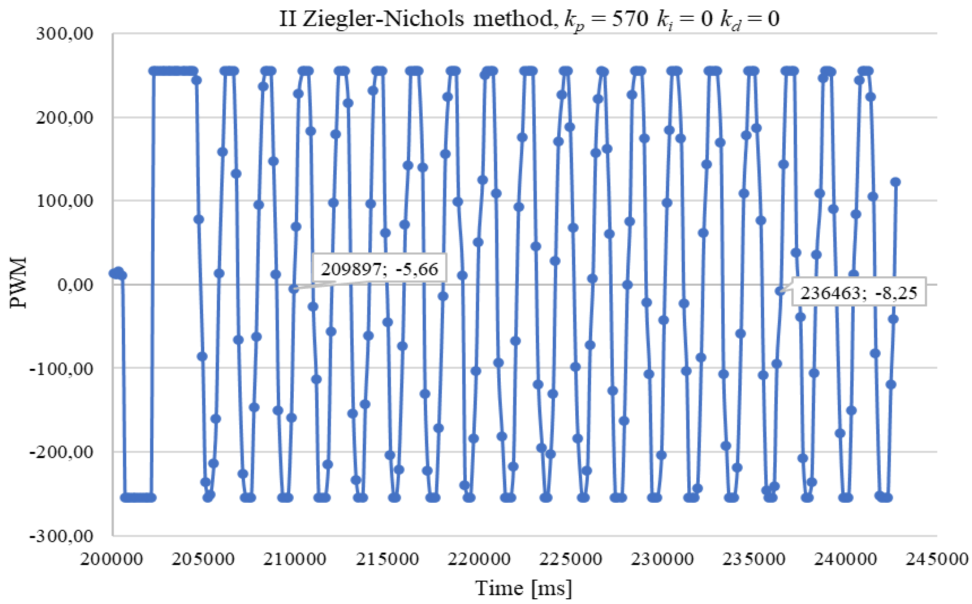


Fig.5. Determination of regulator settings II Ziegler-Nichols method without the dead zone correction

Two samples of the signal as close as possible to the intersection with the x-axis are marked on the graph. To determine the time of the oscillation, the number of periods and times were read: T_1 = 209897 ms, T_2 = 236463 ms, n = 13 and the oscillation time was calculated:

$$T_{osc} = \frac{T_2 - T_1}{n} = 2.043s. \tag{13}$$

Then, the controller settings were determined, the values of which are listed in Table 4. The gains of the PI controller were also determined.

Table 4. Regulator settings with the dead zone turned off

	K_p	T_I, K_I
PI	$0.45 \cdot k_r = 256.5$	$T_I = 0.85 \cdot T_{osc} = 1.73655$ $K_I = K_P/T_I = 147.7$

Figure 6 shows the response of the system without the dead zone correction.

In the system in which the dead zone correction has not been introduced, oscillations at the level of 7% of the set value are visible. They result from the summation of the integral term error. The control system generates a control signal, but for a long time the control object does not react, and the error adds up. It should be noted that an anti-windup system has been implemented in the system, which prevents the system from oversaturation, which eliminated even greater oscillations.

In the next study, the regulator settings determined using the II Ziegler-Nichols method and the manually selected settings were compared. Based on trial and error as well as analysing the data on the graph on an ongoing basis, the following regulator gains were selected: K_P = 180, K_I = 38 and K_D = 10. For analysis, the graphs were compared with the step response in both configurations. The operation of the regulators at a constant disturbance was compared.

Fig. 7a shows a comparison of the response of the system with the settings of the PID controller selected by the II Ziegler-Nichols method and selected manually. Based on the waveform obtained, it can be concluded that the settings selected manually resulted in greater overshoot and reduced oscillations around the set value. Fig. 7b shows a comparison of the system control with a constant disturbance. You can notice a temporary greater overshoot at the set value jump in the case of own settings, but smaller deviations in the steady state.

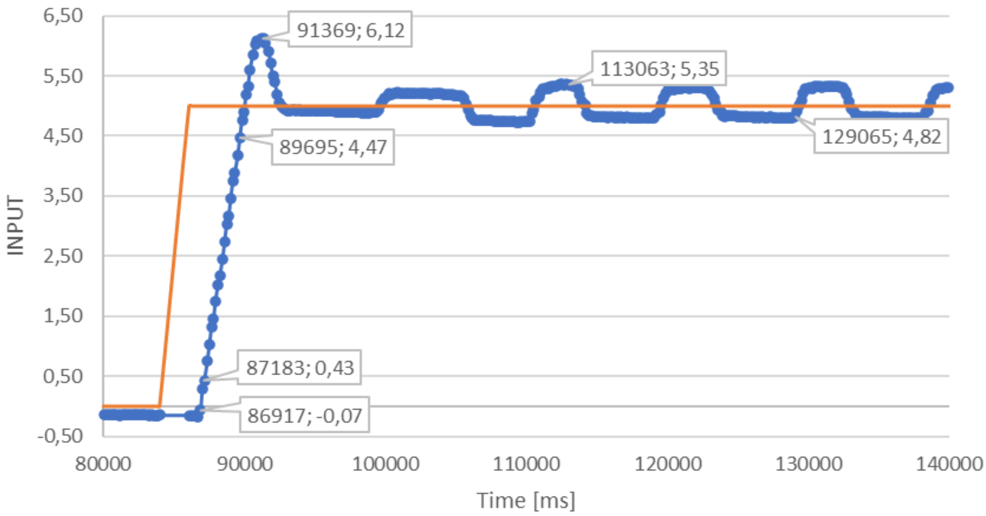


Fig.6. Response of the system without the dead zone correction – PI controller

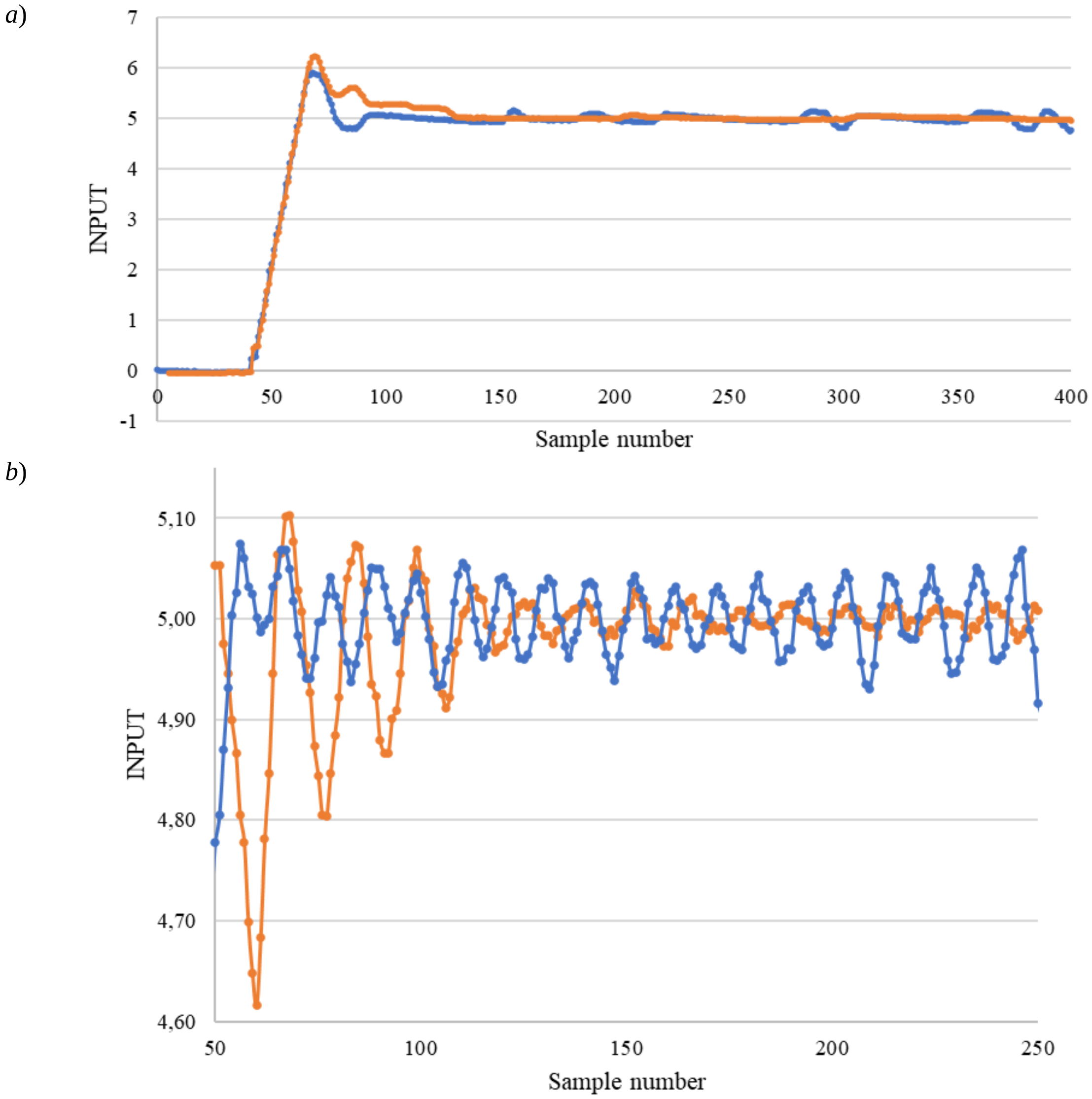


Fig.7. Comparison of the system's response with settings selected by II Ziegler-Nichols method and manually selected - PID controller

Selection of settings using the II Ziegler-Nichols method brought particularly satisfactory results. However, there are no universal methods, the regulator settings depend on the expectations of the system. Depending on which properties will be critical, speed of response, maximum overshoot, steady-state error, ability to track setpoint, the parameters will differ from each other.

4. CONCLUSIONS

The work presents the method of building the system, the selection of elements and the principle of operation of the system. Then, tests were conducted on the system comparing the operation of individual elements of the PID controller, the tuning of the controller was conducted using the II Ziegler-Nichols method and the response of the system to a step change of the set value was compared. Control quality indices, maximum overshoot, rise time and control time were determined for the response analysis. In addition, the responses of different controller configurations during constant disturbances were compared to assess how the controller behaves when it must work in the follow-up mode. The disturbance was adding water to the tank at a constant flow rate. In the next stage, it was checked what effect the exclusion of the dead zone correction of electric motors had on the system in the case of the PID controller. Based on the obtained results, it can be concluded that the P-type controller is a particularly good controller, if there are no permanent disturbances in the system and the controller does not have to operate in follow-up mode. The PI and PID controllers worked very well both with the change of the set point and with constant disturbances. Deviations from the set value did not exceed 2%, which in fact translates into deviations of 2 ml of water. However, if the dead zone has not been corrected, significant oscillations occurred in the steady state. Therefore, it can be concluded that in such systems, when there is a large dead zone of the actuator, it should be levelled in the controller by appropriate control, which significantly affects to improve the performance of the regulator. The constructed laboratory stand is an example of a complex automatic control, which was built from non-industrial elements, while the principle of operation is no different from such. Using simple and cheap elements, a system was presented, which in industrial realities is built on ready-made regulators and PLC controllers. On the other hand, the built stand allows for modifications, improvements and conducting research that would not be possible on industrial systems.

Hardware and software improvements can be introduced into the system. One of them could be the introduction of several safeguards, e.g., adding a liquid level sensor and programming it in Arduino so that there is no possibility of liquid overflowing in the tank, which could happen if too high a set value is entered or incorrect regulator settings. Another improvement results from the need to connect the system to a computer when you need to draw graphs. It is possible to introduce complete autonomy of the system by using, for example, the Raspberry Pi computer platform. Instead of connecting the Arduino to a PC, it could be connected to a Raspberry with Telemetry Viewer installed. It would also be necessary to connect the Monitor via the HDMI port. Thanks to this solution, it would be possible to use the system and draw graphs on the screen on an ongoing basis, without connecting a PC. These works will be the subject of further research.

5. REFERENCES

1. Asraf, M., Dalila, K. Hakim, M. (2017) *Development of Experimental Simulator via Arduino-based PID Temperature Control System using LabVIEW*, Journal of Telecommunication, Electronic and Computer Engineering, 9, 53-57.

2. Kuźnik, J. (2006) *Regulators and control systems*, Silesian University of Technology Publishing House, Gliwice (In Polish).

3. Lis, S., Tomasik, M., Korenko, M. Findura, P. (2018) *The use of a PI controller with a modified structure in the process of controlling an induction furnace*, Przegląd elektrotechniczny, Vol. 12, 138-141 (In Polish).

4. Moćko, W. (2017) *Experimental methodology for evaluating the sensitivity of alloys to strain rate*, Transport Samochodowy, 2, 69-86 (In Polish).

5. Płaczek, M., Piszczek, Ł. (2018) *Testing of an industrial robot's accuracy and repeatability in off and online environment*, Eksploatacja i Niezawodnosc – Maintenance and Reliability, 20(3): 455–464.

6. Płaczek M., Piszczek Ł. (2018) *The method for tests of emergency stopping of industrial robots. Modelling in Engineering*, Modelowanie Inżynierskie, 38(69), 60-67.

7. Płaczek M., Nowak P. (2019) *Examination of the pneumatic slide valve due to the safety of pneumatic drives*, International Journal of Modern Manufacturing Technologies, 11(2), 80-85.

8. Tomera, R. (2010) *Regulator PID*, Automatyka, Elektryka, Zakłócenia, 1(2), 89-93 (In Polish).

9. Pomiar siedzeń 7 - Verma, S., Tieman, C., Chendalva, M. K., Fortune, D.D. (1999) *Method and apparatus for measuring seat loading by strain gauge*, Patent no. USOO5942695A.

10. Wróbel, A. (2020) *Pneumatic elements of machines tools*, International Journal of Modern Manufacturing Technologies, 12(3), 137-142.

11. Zakrzewski, J. (2001) *Basics of Automation Laboratory and a selection of examples for exercises*, Silesian University of Technology Publishing House, Gliwice (In Polish).

Received: July 30, 2022 / Accepted: December 15, 2022 / Paper available online: December 20, 2022