Ministry of Education of the Republic of Belarus
Educational Institution "Euphrosyne Polotskaya State University of Polotsk"

____________ As a manuscript

Al Ajami Wissam Khaled

Study of a control algorithm based on fuzzy logic and its use at electric power facilities

Master′s thesis

specialty 1-43 80 01 Electrical Power Industry and Electrical Engineering

Research advisor
Pitolin Uladzimir
Technical Sciences Associate Professor

Accepted for defense
___ ____________2024
Head of the Department _________________,
(Name of the department)
____
(academic degree, academic title)
_____________ Full name
(signature)

Novopolotsk, 2024

Note-Italicized explanatory text is not typed

CONTENT

GENERAL CHARACTERISTICS OF THE WORK

KEY WORDS : PID-CONTROLLER, DELAY, CONTROL INSTABILITY, NEURAL NETWORKS, PERCEPTRON.

The aim of the study: analyze existing process control algorithms with a delay; development of a regulator based on modern neural technologies, which makes it possible to ensure stability in the regulation of such processes.

Object of the research: the regulator of an object characterized by a large transport delay.

Subject of the research: Let us define the following objectives of the dissertation research:

- analyzing the operating parameters of the PID control algorithm and studying the reasons for their instability;
- study of existing methods for correcting the characteristics of regulators to improve the quality of regulation of objects with a large delay;
- development of neural networks for predictive assessment of regulatory impacts and construction of a regulator based on a self-learning perceptron;
- experimental testing and assessment of the reliability of the research results obtained.

Research methods: Literature analysis, development of a software model of the regulator, research method at the bench and on site.

Realm of the possible practical application: he dissertation explores the topic of modern automatic control theory, namely the creation of controllers with fuzzy logic, in order to ensure the regulation of objects with a large delay, for which the use of PID laws is unacceptable.

Such objects in particular include heating systems for industrial and residential buildings. Of course, this topic is relevant and promising. It fully fits into the state program “Comfortable Housing and Favorable Environment” for 2021-2025.

Introduction

Automation is the process of performing management and control functions using means and methods of regulating the technological process without human intervention. The quality of control of these processes is completely determined by the control algorithms used. The most optimal of them is the PID algorithm. A PID control unit is available in any industrial controller. A large number of methods for setting its parameters, including automatic ones, have been developed. The quality of process control is ensured by setting these parameters.

However, not all so simple. Any, even the most perfect regulator, has its limits of use, determined by the stability margin determined by the properties of the object of regulation itself, namely the amount of transport delay and inertia.

The relevance of the study is related to the widespread use of technological complexes with transport delays. Lag is, as a rule, a delay associated with the movement of a product or energy along long communication lines, for example, the movement of a substance in a production process at a certain speed without changing its properties, as well as the increased inertia of the control object itself. All this worsens the quality indicators of transient processes, and sometimes even makes it impossible to use the PID control algorithm, for example, in heating or hot water supply systems.

Therefore, the creation of tools or algorithms that increase the stability and quality of operation of an automated system is a very relevant, important and complex task.

If we turn to the experience of living nature, we can see that the absence of complex algorithms for controlling living organisms does not in the least impair their behavior and controllability. Living organisms do not perform complex calculations of the magnitude of the impact, but use ready-made solutions based on their experience (knowledge base)!

Everything is quite simple: a synaptic data collection system and neural networks for subsequent processing of information by a living organism, which, based on previous experience, allows you to predict the results of possible actions and select the most optimal ones, and if they fail, remember and correct control errors.

A mathematical or computer model of such a system, developed by Frank Rosenblatt in 1957 [1], is called a perceptron. The perceptron consists of three types of elements, namely: signals received from sensors (elements of the first type) are transmitted to associative elements that make up the knowledge base of the processing and comparison algorithm, and then to reacting elements. Thus, perceptrons allow you to create a set of “associations” between input signals and the desired output response.

Perceptron training consists of correcting “associations” (knowledge base) when the results of the influence do not coincide with the expected result and is performed constantly, as an additional function of the perceptron.

The heating system of a 9-story building, characterized by a very large delay time and inertia (transport delay time - 12 minutes), was chosen as an object of regulation in this work.

To form an object control algorithm based on neural technologies, we will build a 2-layer perceptron:

- On the first layer, the input signal is associated with the current task, determined by the control tasks (ambient air temperature), and the mismatch value is determined.

- At the second layer, according to the “association” available in the knowledge base, the magnitude of the regulatory impact is determined, determined by the design of a specific regulatory body, the existing mismatch and the ambient temperature (the opening value of the throttle valve for direct network water supply).

- The result is assessed after the regulation time has expired (12 minutes). If there are discrepancies, the “association” database is corrected (this is how the perceptron is “trained”).

The purpose of the research is to develop an algorithm for controlling technological facilities in the electric power industry with a large delay, using neural networks to improve the stability and quality of regulation.

CHAPTER 1
AUTOMATIC REGULATION OF TECHNOLOGICAL PROCESSES

1.1 PID Controllers, Their Advantages and Disadvantages

An automatic regulator is an automation tool that receives, amplify and converts the deviation signal of the controlled quantity and purposefully influences the controlled object. It ensures that the specified value of the controlled variable is maintained or its value changes according to a given law.

Based on their design characteristics, automatic regulators are divided into:

- Hardware-based;
- Instrument-based;
- Aggregate-based;
- Modular (component-based)

Hardware-type regulators are structurally a technical device that operates in conjunction with a primary measuring transducer. Hardware automatic controllers operate independently (in parallel) from the means of measuring a given process parameter.

Instrument-based controllers only operate in conjunction with a secondary measuring instrument. Instrument-based controllers do not have a direct connection to the primary measuring transducer.

Typically, the functional diagram of the regulator is built along a closed loop in order to eliminate the deviation of the controlled variable from the set value (Figure 1.1).

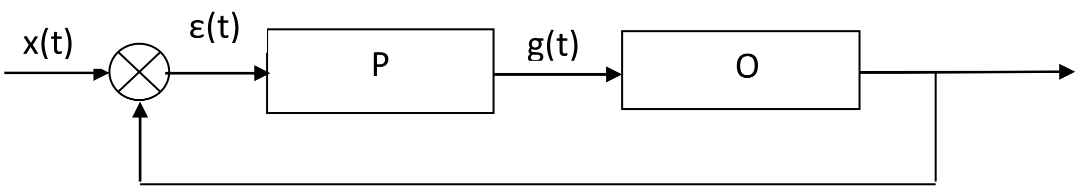


Figure 1.1- Block diagram of a simple regulator

The measuring device D has a setting device on which the required set value of the controlled variable x(t) is manually set. The specified value x(t) in the device is compared with the actual value y(t) of the controlled variable, determined by the position of the moving measuring system of the device, and the difference ε(t) = y(t) – x(t) is fed to the input of the controller P, which generates the control influence of g(t) on object O.

In some cases, the instrument-based control device itself is housed in the same enclosure as the secondary measuring instrument. Thus, instrument-based controllers are connected in series with the secondary measuring instrument.

The advantage of instrument-based controllers is that in this case, there is no need to install additional primary measuring transducers or lay communication lines from them to the controllers. Their disadvantage is lower dynamic performance and reliability compared to similar parameters of hardware-based controllers.

Automatic regulators, built according to the aggregate (block) principle, consist of separate unified blocks that perform specific functions. The input and output signals from these blocks are unified. This allows you to design automatic regulators for various functional purposes from blocks.

Automatic controllers built on the modular (component-based) principle consists of individual modules (elements) that perform simple operations. The input and output signals of these modules are standardized. This allows, similar to the use of aggregate controllers, to assemble automatic controllers with different functional purposes.

Depending on the source of energy used, automatic controllers are divided into two types: direct-acting controllers and reverse-acting controllers.

Regulators of this type are widely used in local control systems for industrial equipment, centralized process control systems, and in robotics. The devices allow you to quickly return an adjustable parameter to an acceptable range, accurately maintain the value and quickly respond to disturbing influences. Thus, no external energy is supplied to the automatic “object-regulator” system.

In reverse-acting automatic controllers, energy from an external source is supplied to operate the controller and actuate its executive mechanism. Depending on the type of energy used, reverse-acting controllers are divided into: electric (electromechanical, electronic), pneumatic, hydraulic, and combined (electropneumatic, electrohydraulic, etc.) controllers.

The designer's task is to select a controller type that would provide the desired control quality with minimal cost and maximum reliability. The developer can choose between relay, continuous, or discrete (digital) types of controllers. To select the controller type and determine its settings, the following information is required:

- Static and dynamic characteristics of the controlled object.
- Requirements for the quality of the control process.
- Performance indicators for control quality in the case of standard controllers.

- Nature of disturbances acting on the control process.

In recent years, the practice of controller design has established the proportional-integral-derivative (PID) controller as the most optimal solution. A PID controller is a device used for automatic control of one or multiple parameters within a specified range. PID controllers are universal devices capable of implementing various control laws [1-4].

PID controllers can take into account the actual value, the set value, the difference in values and the rate of change of the controlled characteristics.

Regulators of this type are widely used in local control systems for industrial equipment, centralized process control systems, and in robotics. The devices allow you to quickly return an adjustable parameter to an acceptable range, accurately maintain the value and quickly respond to disturbing influences.

A proportional controller used in a system typically provides an output value that is proportional to the current error. The controller compares the desired or setpoint value with the actual or feedback value of the process. The resulting error is multiplied by the proportional gain to determine the output value. A higher proportional gain leads to a smaller output value for the same error. If the error is zero, the output value will also be zero. Proportional control is typically used in low-inertia systems with a high gain.

To tune a proportional controller, initially set the proportional gain to its maximum value. During this process, the output value will be minimized. Once the measured value is stabilized, set the desired value and gradually decrease the proportional gain, which will reduce the control error. When the system starts to exhibit periodic oscillations, adjust the proportional gain so that the control error and system oscillations are minimized.

The integral controller cannot predict the future behavior of the error. To address this issue, a differential controller is introduced into the system. The output value of the controller depends on the rate of change of the error over time, multiplied by the derivative constant. When using a PID controller, the output value is a sum of three components: proportional, integral, and derivative. The derivative component provides information about the rate of change of the error. The larger the derivative time constant, the stronger the system's response to disturbances.Usually, a PID controller is applied in inertial systems where disturbances from the measurement channel are relatively small. The advantages of this type of control include fast response to reach the desired operating state, precise maintenance of the required temperature, and quick response to disturbances.

The PID controller is one of the key elements of the process control system (ACS) and is designed to regulate technical parameters according to the proportional-integral-derivative law.

The algorithm of the PID controller can be represented as follows [1-5]. The PID controller has three main control effects. The proportional (P) action provides a change in the input signal (controlled variable) that is directly proportional to the error signal. The integral (I) action provides a change in the input signal that is proportional to the integral of the error, and its main goal is to eliminate the error. Meanwhile, the derivative (D) action is used to speed up the response or stabilize the system, providing a change in the input signal that is proportional to the derivative of the error signal. The overall output of the controller is the sum of these three components [4]. The general form of the PID controller is shown below in equation (1.1).

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

(1.1)

where u(t) and e(t) denote the control and error signals, respectively, and the integral time of the proportional gain (Kp) (Ti) and derivative time (Td) are the parameters to be adjusted.

The block diagram of a closed-loop PID controller is shown in Fig.1.2.

Figure 1.2 - Block diagram of a closed-loop PID controller

In Figure 1.2, the following notations are introduced:

P - the controlled object.

R - the controller.

y - the output signal (controlled variable).

u - the control signal, the output of the controller.

e - the error signal (deviation between the setpoint and the output signal).

r - the input reference (setpoint).

Using Laplace transform and assuming zero initial conditions, the transfer function of the PID controller can be represented in the following form:

$$U(s) = K_p \left(1 + \frac{1}{TiS} + T_d * S \right) = K_P + K_i \frac{1}{S} + K_d * Sq \qquad (1.2)$$

where s is the complex frequency.

Figure 1.3 shows the structure of a PID controller in parallel form.

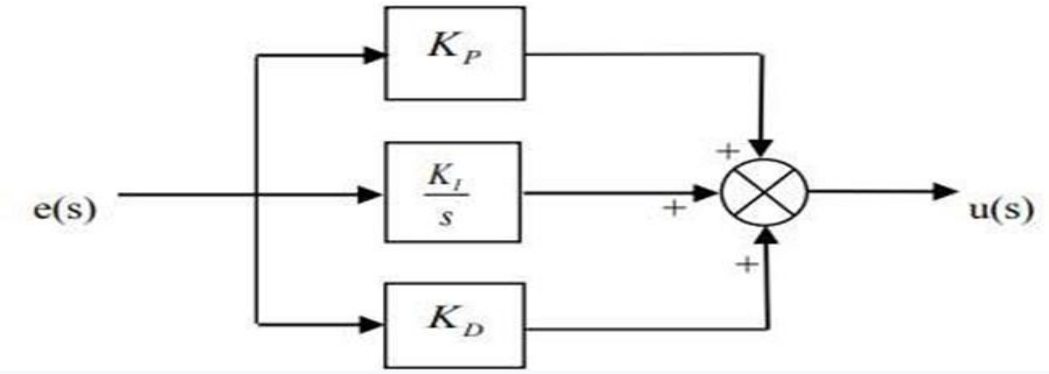
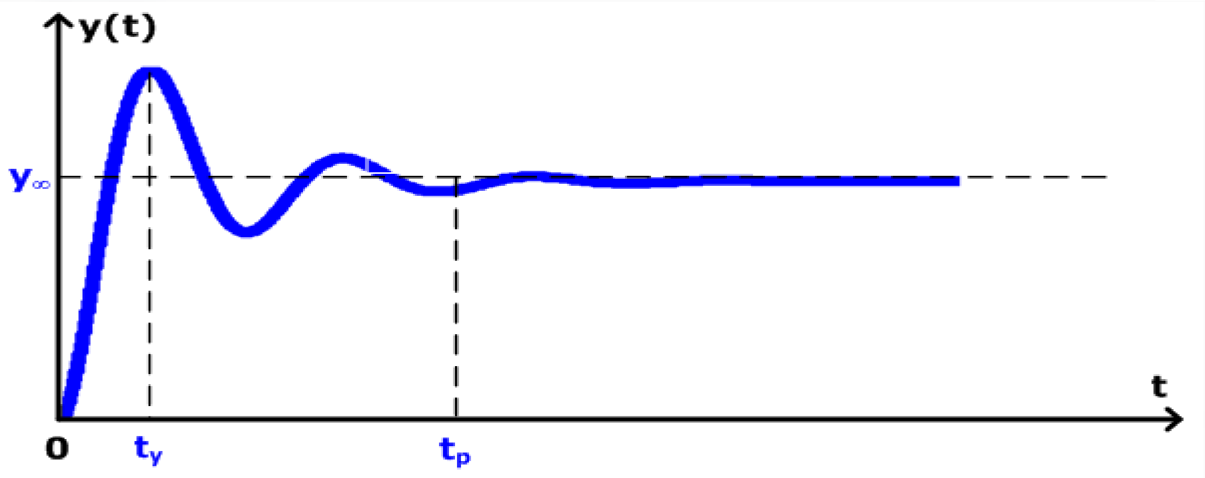


Figure 1.3 - PID controller structure

Where Kp are the gains of the proportional components of the controller, Td are the gains of the differentiating components of the controller, Ti are the gains of the integrating components of the controller, Td is the derivative time, and Ti is the integral time. The integrating component controller improves the transient process by adding zero to the transfer function of the open system.The integrator eliminates the error by increasing the system type with an additional pole at the origin. In general, Kp will have the effect of reducing the rise time and also reducing the error, but the transient error can never be eliminated. The integral gain Ki can be used to eliminate the steady state error, but this will make the transient process worse.

Figure 1.4 shows the usual transient process of changing the characteristics of an object with a step disturbance at the input.

In order to obtain the optimal shape of the transition graph of the process, it is necessary to configure the PID controller. Setting the parameters of the PID controller during operation can be performed using various methods [8, 9]. The most optimal type of transient process is shown in Figure 1.5. Such a PID controller provides the highest quality control of the technological process.



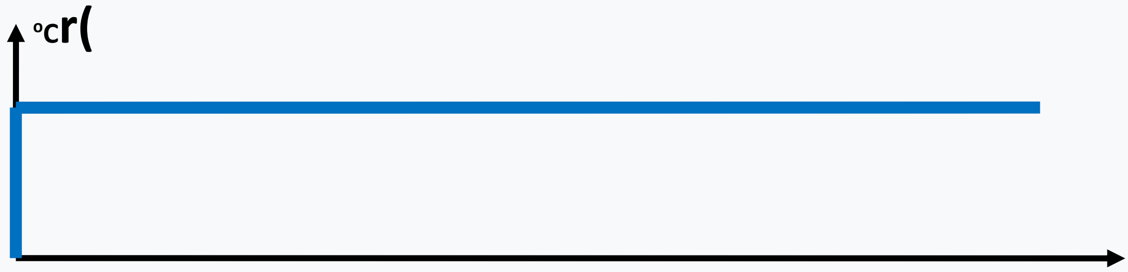


Figure 1.4 - Step response of the controlled object

The transfer function of a control object that has a smooth, non-oscillatory transient response, presented in Figure 1.5, is quite simply achieved using various calculation or graphical methods for tuning the PID controller [5, 6].

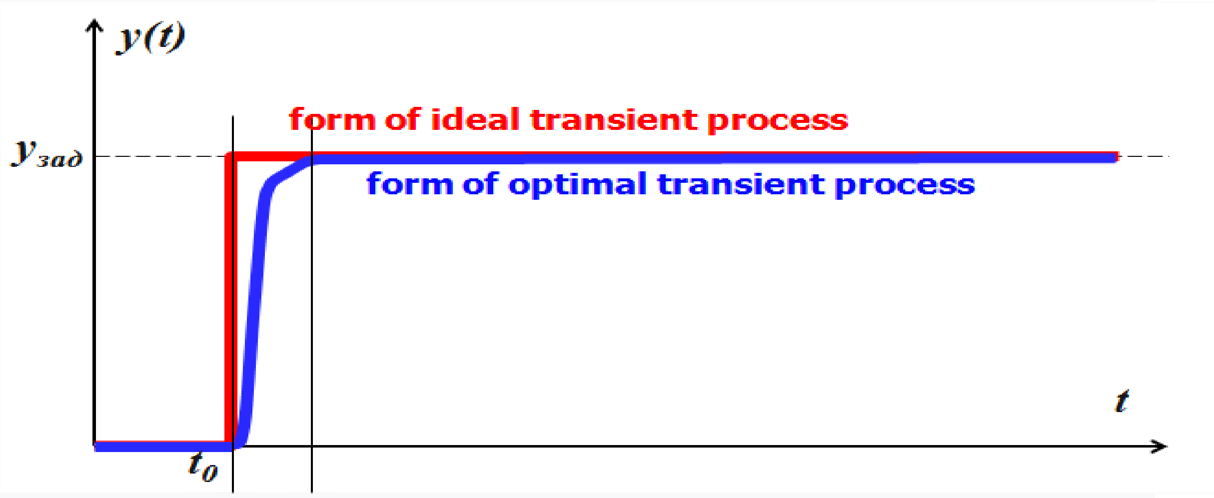


Figure 1.5 - Optimal transient response of the control object under stepwise input action

1.2 The Concept of Transport Delay of the Regulated Object

A significant portion of industry involves objects with transport delay. There are numerous examples of such cases. These are primarily found in manufacturing processes where the components of the technological process are sufficiently distant from each other, resulting in the emergence of a delay link within the technological process.

An example of a technological process with a pure transport delay is the process of transporting bulk materials.

In this case, the process begins with immersing the material in a hopper, from where it enters through a dispenser device onto a belt conveyor; as the material moves along the belt conveyor, the material enters the next necessary technological process unit, Figure 1.6. [7].

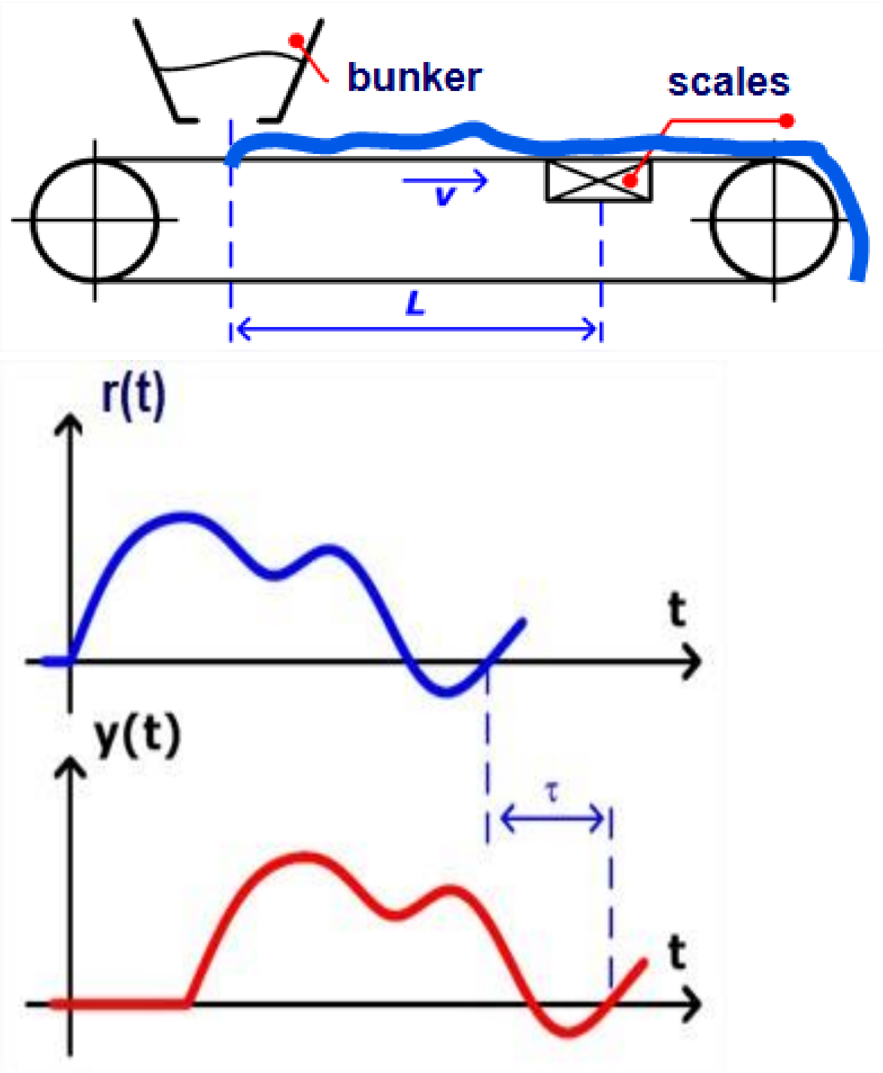


Figure 1.6 - Example of an inertia-free link with pure transport delay τ.

The magnitude of the transport delay can be estimated using formula (1.3):

$$\tau = \frac{L}{V} \qquad (1.3)$$

Typically, all regulated objects have transport delays to one degree or another. In addition to this characteristic, when setting up regulators, one should also take into account the inherent inertia of the regulation process, determined by the value of T.

The characteristics of a real self-leveling control object, characterized by transport delay and inertia, usually look as shown in Figure 1.7.

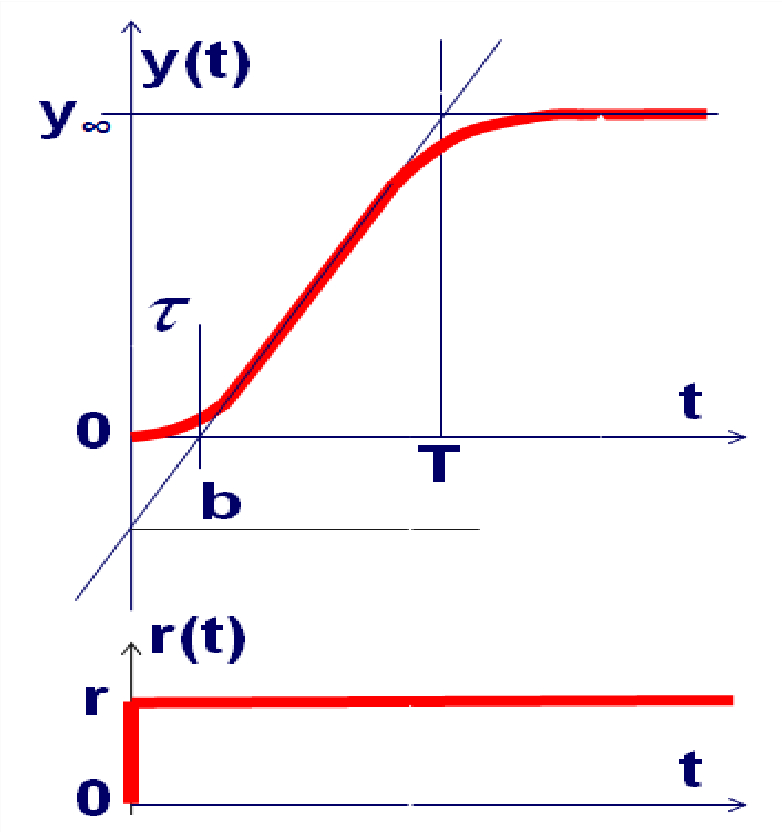


Figure 1.7 - Example of characteristics of a real regulated object**.**

Such an object is characterized by both a **transport delay** (τ) and some **inertia** (T). The transfer function of such an object typically appears as follows:

$$W(s) = \frac{k \cdot e^{-s\tau}}{1+T \cdot s}, \qquad (1.4)$$

where k=Y∞/r is the conversion coefficient of the regulated object.

For ease of calculating optimal coefficients for a PID controller, the regulated object itself is typically characterized by the ratio of **τ/T**.

Thus, with a very small value of this parameter and a finite value of the transport delay, the object is considered very inertial, and with a large value, the object has a large transport delay. In both cases, the use of a PID controller in its classical representation becomes problematic and does not provide stable control of the technological process.

1.3 Instability of PID control during transport delay

To observe the impact of delay on a system, let's model the behavior of two identically tuned systems. The only difference between them will be the presence of a delay element in one of them. Let's denote these systems as (a) and (b) (see 1.8).

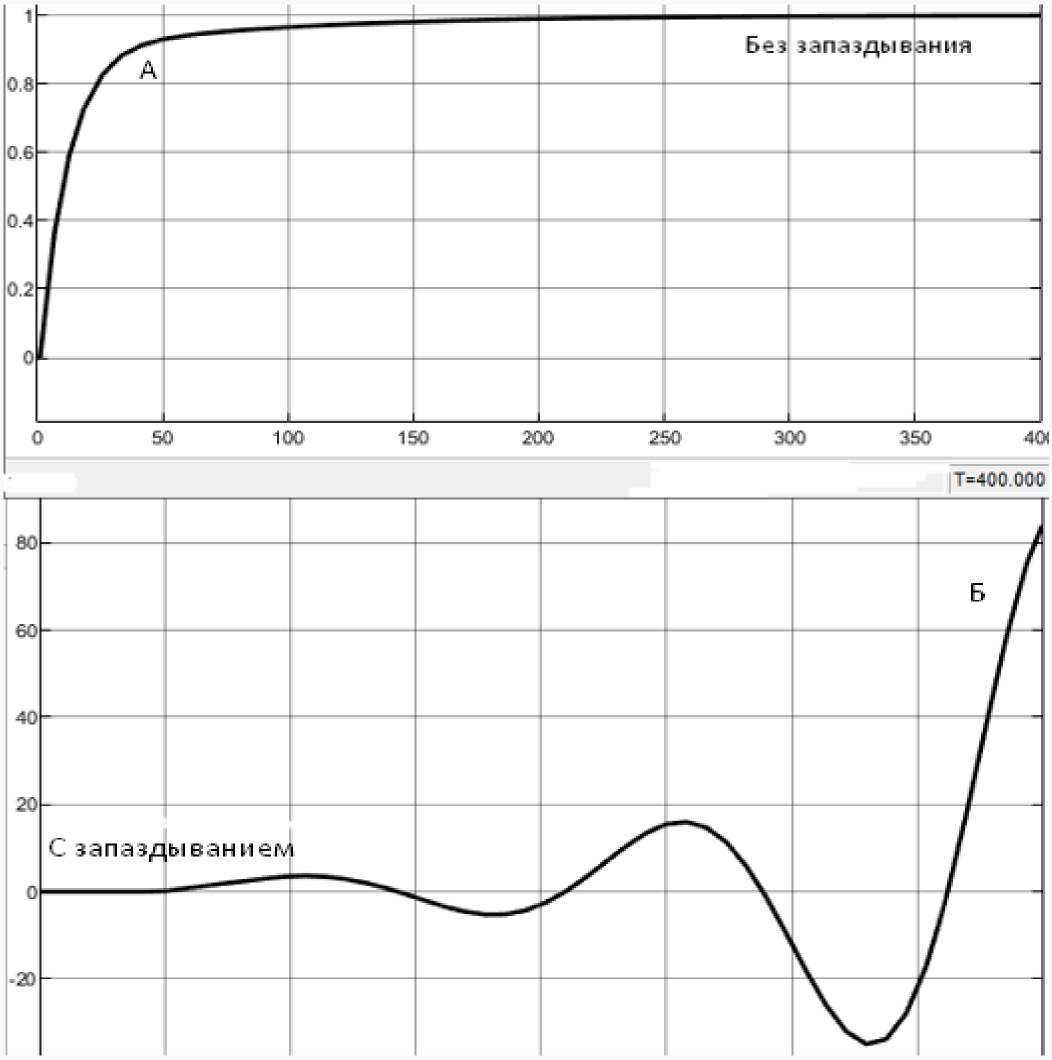


Figure 1.8 - A system without delay containing a PID controller (a), and a system with delay containing a PID controller (b).

From this comparative analysis we can conclude that the delay link has a negative impact on the system. Figure 8b shows where the system becomes unstable.

Automatic control systems (ACS) can have links in which the relationship between the output and input characteristics has the form: $y(t)=u(t-\tau)$, where τ is a constant value called the delay time.

Since these links can reproduce the input value (with some delay), they are called retarded. Figure 1.9 shows a single-circuit structure of an automatic control system with a delay link.

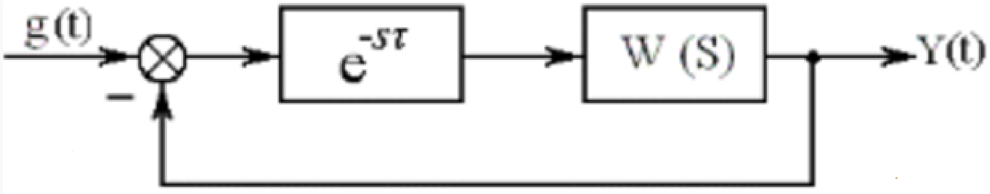


Figure 1.9 - automatic control system with delay

The Transfer function of the delay link:

$$W_{зап}(s) = e^{-\tau s} \qquad (1.5)$$

Examples where time delays play a role can be found in technological processes involving the movement of production goods or materials from one point to another. In control systems, these processes can be described using differential-difference equations. Control systems that incorporate at least one time delay are characterized as systems with **time-delay elements**. Thanks to transfer functions for systems with lumped parameters and high-order systems with lumped parameters, it becomes possible to approximate the transfer functions of complex systems. The structure with a time-delay element in the forward path can be represented as shown in Figure 1.9.

The transfer function for an **open-loop system with a time-delay element** can be represented as follows:

$$W_\tau(s) = W_{зап}(s)W(s) = \frac{R(s)}{Q(s)}e^{-\tau s}, \qquad (1.6)$$

where $W(s)$=$R(s)$/$Q(s)$ is a fractionally rational function of the operator S (without a delay link, if there are many delay links, then they add up).

The transfer function of a closed-loop system (if there is a delay link in the direct chain) can be represented as:

$$W_{gx} = \frac{W_\tau(s)}{1+W_\tau(s)} = \frac{R(s)e^{-\tau s}}{Q(s)+R(s)e^{-\tau s}} = \frac{R_\tau(s)}{D_\tau(s)} \qquad (1.7)$$

From the expression mentioned above, we can conclude that a **time-delay element** affects the characteristic equation :

$$D_\tau(s) = Q(s) = R(s)e^{-\tau s} \qquad (1.8)$$

According to the above-mentioned characteristic expression, it can be argued that it is transcendental (is an expression of “infinite degree”) since it has an infinite number of roots.

Due to the fact that:

$$e^{-\tau s} = 1 - \tau s + \frac{s^2\tau^2}{2!} - \frac{s^3\tau^3}{2!} + \cdots$$

For the stability of systems with delay, it is necessary that the roots found are left-handed, but, unfortunately, detecting the roots of the equation is problematic; therefore, when detecting them, stability criteria are used, for example, the Nyquist or Mikhailov stability criteria.

To obtain an expression for Mikhailov's hodograph, it is necessary to substitute s=jω into the characteristic equation:

$$D_\tau\left(j\omega, e^{-j\omega\tau}\right) = Q(j\omega) + R(j\omega)e^{-j\omega\tau} \qquad (1.9)$$

With the help of the Mikhailov criterion, unfortunately, the graph of the Mikhailov curve becomes difficult to understand and the formulation of this criterion is not so simple due to the presence of $e^{-j\omega\tau}$, and then a more convenient criterion is the Nyquist criterion.

Information aboutwhether the system is stable can be obtained based on the amplitude-phase characteristic (APC) $W_\tau(j\omega)$ of the system relative to the point (-1; j0). The frequency transfer function $W_\tau(j\omega)$ of a system with a delay is found by substituting s= jω into expression (1.6).

Where can we say that:

$$W_\tau(j\omega) = W_{зап}(j\omega)e^{-j\omega\tau} = A(\omega)e^{-s\omega\tau}e^{j\varphi(\omega)} = A(\omega)e^{j\varphi_\tau(\omega)} \qquad (1.10)$$

where W(jω)=U(ω)+jV(ω) is the AFC of the system without taking into account the delay;

$A(\omega)=|W(j\omega)|=\sqrt{U^2(\omega) + V^2(\omega)}$amplitude-frequency response;

$\varphi(\omega)=Arctg V(\omega)/U(\omega)$ - phase-frequency response (PFC) of an open-loop system without taking into account delay;

$\varphi\tau(\omega)=\varphi(\omega)-\omega\tau$ - phase-frequency characteristic (PFC) of an open-loop system with delay.

From expression (1.10) and $\varphi\tau(\omega)=\varphi(\omega)-\omega\tau$ it is clear that the presence of a delay link does not change the modulus A(ω) of the AFC of the system $W_\tau(j\omega)$, but only brings an additional negative phase shift ωτ proportional to the frequency, at This proportionality coefficient is the delay time τ.

Knowing the AFC W(jω) of a system without delay, it is easy to construct the AFC $W_\tau(j\omega)$ of a system with delay. To do this, each module $A(\omega j)$ of the AFC vector W(jω) must be rotated by an angle $\omega_j\tau$ clockwise. With increasing frequency ω, the angle $\omega\tau$ will quickly increase, and the modulus A(ω) usually decreases, therefore the amplitude-phase characteristic $W_\tau(j\omega)$ of an open-loop system with delay has the form of a spiral twisting around the origin, as shown in Figure 1.10.

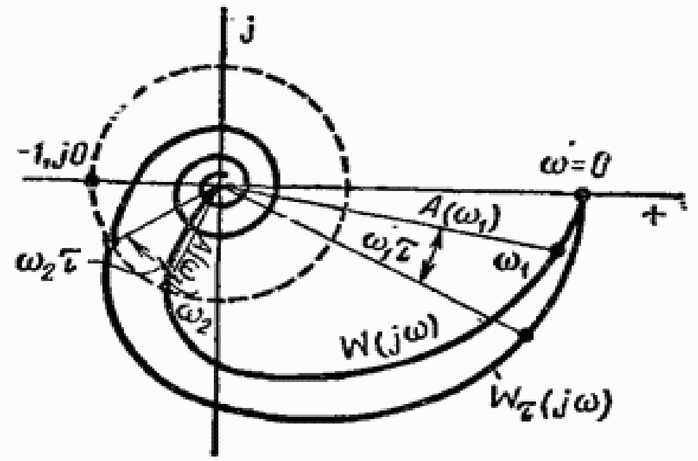


Figure 1.10 - AFC W_τ(jω)

The “spiraling” of the amplitude-phase frequency response (AFR) due to the presence of an additional phase shift $\omega\tau$ generally deteriorates the stability of the system, as the entire AFR approaches the critical point (-1, j0)

However, sometimes introducing a constant time delay can improve stability conditions when dealing with a complex amplitude-phase frequency response (AFR) represented by W(jω). By adjusting the delay parameter τ, it’s possible to find a value where the system approaches the stability boundary, and the characteristic $W\tau$(jω) passes through the point (-1, j0). The delay time τкр and the corresponding critical frequency value ωкр at which $W\tau$(jω) intersects the point (-1, j0) are referred to as critical parameters. In such critical cases, specific conditions apply

$$W_\tau(j\omega_{кр}) = W(j\omega_{кр})e^{-j\omega_{кр}\tau_{кр}} = A(\omega_{кр})e^{j[\varphi(\omega_{кр})-\omega_{кр}\tau_{кр}]} = -1 \quad (1.11)$$

The separate expressions for the amplitude and phase of the vector $W\tau(j\omega_{кр})$ can be derived as follows:

$$\mathrm{A}(\omega_{кр}) = |\mathrm{W}(\mathrm{j}\omega_{кр})| = 1 \quad (1.12)$$

$$\varphi_\tau(\omega_{кр}) = \varphi(\omega_{кр}) - \omega_{кр}\tau_{кр} = -\pi(2j+1), \quad \text{where } j=0, 1, 2, \ldots$$

From $A(\omega_{кр})=|W\tau(j\omega_{кр})|=1$ you can first find ω ckpand then from $\varphi_\tau(\omega_{кр})=\varphi(\omega_{кр})-\omega_{кр}\tau_{кр}=-\pi(2j+1)$ find τ_{kp}, that is

$$\tau_{кр} = \frac{\varphi(\omega_{кр})+\pi(2j+1)}{\omega_{кр}} = \frac{\pi+\varphi(\omega_{кр})}{\omega_{кр}} + \frac{2\pi}{\omega_{кр}}j \quad (1.13)$$

Where $\varphi(\omega_{кр}) = \pi + Arctg\frac{V(\omega_{кр})}{U(\omega_{кр})}$ phase stability margin.

When dealing with a complex expression for W(jω), determining the **critical time delay** is often easier to visualize graphically. The condition A(ωкр) = |$W\tau$(jωкр)| = 1 is established by the intersection of the Nyquist plot (or polar plot) of W(jω) with a unit-radius circle centered at the origin (Figure 1.11)

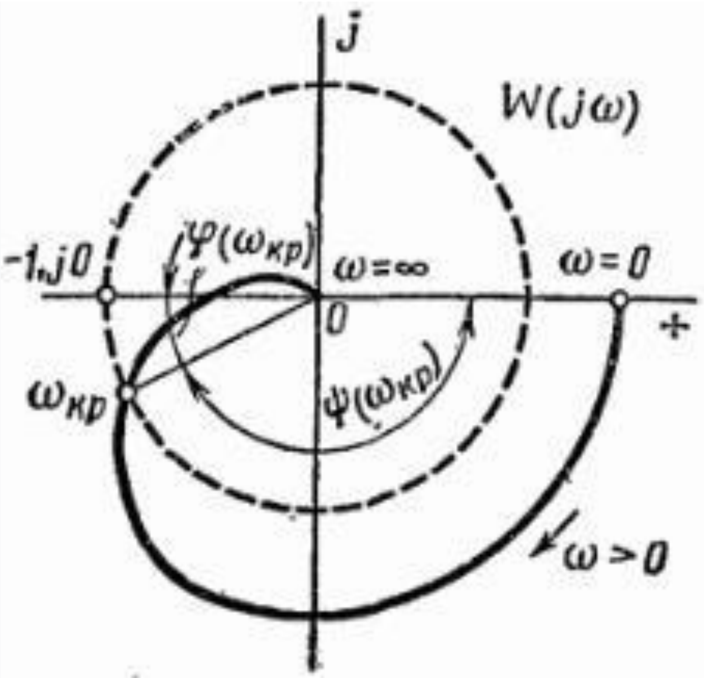


Figure 1.11 - Hodograph W(jω)

The intersection point simultaneously determines both ωкр and the angle φ(ωкр). When there are multiple intersections of the Nyquist plot (or polar plot) of W(jω) with the unit-radius circle (for example, at ω1кр, ω2кр, and ω3кр as shown in Figure 12), the system will have multiple critical time delays

$$\tau_{1\kappa p} = \frac{\varphi(\omega_{1\kappa p})}{\omega_{1\kappa p}}; \quad \tau_{2\kappa p} = \frac{\varphi(\omega_{2\kappa p})}{\omega_{2\kappa p}}; \quad \tau_{3\kappa p} = \frac{\varphi(\omega_{3\kappa p})}{\omega_{3\kappa p}}$$

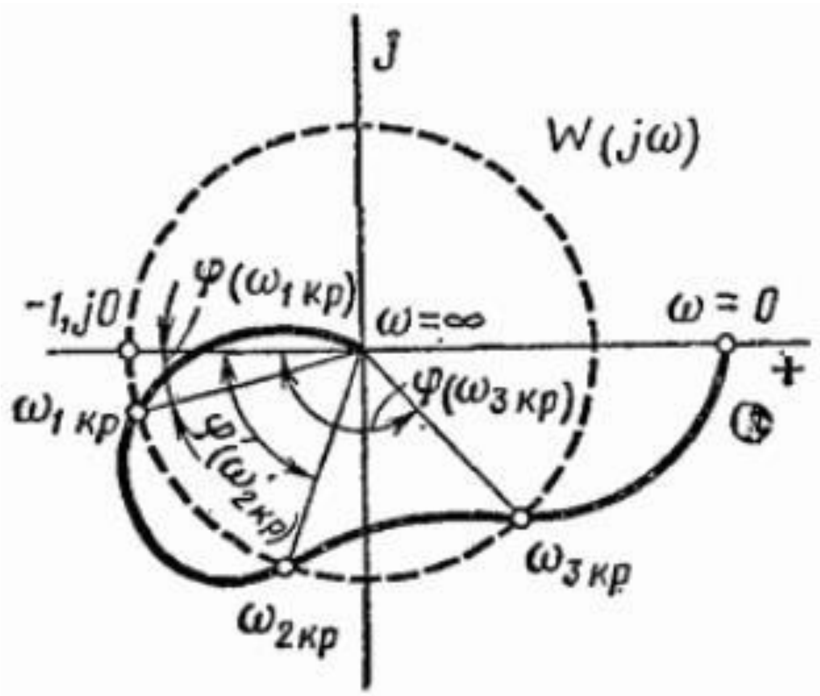


Figure 1.12 - Hodograph W(jω) at several critical boundary delay times.

In this case, the minimum lag time $\tau_{кр\ min}$=$\tau_{1кр}$.

The system will be stable at τ<$\tau_{1кр}$ as well as $\tau_{2кр}$< τ<$\tau_{3кр}$

The system will be unstable at $\tau_{1кр}$< τ<$\tau_{2кр}$, as well as τ>$\tau_{3кр}$.

The alternation of the regions of stability and instability of the system observed in this case at a continuous change τ is a distinctive feature of systems with a delay

Very often, in order to increase the speed and accuracy of the system, the lag time τ is reduced, so the stability criterion is formulated only for the minimum lag time. The SPG will be stable if the lag time τ is less than the minimum critical lag time: τ<$\tau_{кр\ min}$.

It is worth noting that the critical lag time is found even when logarithmic amplitude-frequency (LAF) characteristics and FRF characteristics are used, but then the abscissa axis is represented instead of a circle of the unit radius, and the LAFC of the system coincides with the LAFC of the lag system without delay. The phase shift is determined from the expression φτ(ω)=φ(ω)−ωτ. The points of intersection of the LLAC with the abscissa axis determine the critical frequencies ωjkr, and the phase reserves (taking into account the multiplicity) assigned to the corresponding critical frequencies determine the critical lag times τ_{jcr}.

CHAPTER 2
OPTIMIZATION OF ALGORITHMS FOR REGULATING OBJECTS WITH DELAY

2.1 Smith predictor.

Time delays are a common occurrence in many industrial processes, and they introduce complexities related to process controlPID controllers often prove ineffective for managing such processes because maintaining stability in a closed-loop system requires meticulous tuning of coefficients .

To determine and account for transport delay time Otto J.M. Smith in 1957 introduced the PID delay compensator, which is now widely known as the Smith predictor [10, 11].

The main meaning of the predictor's operation is that after implementing the control action of the PID controller on the object, it switches to the internal loop and no longer affects the object during the transport delay time of the object.

The controller incorporates a process model, allowing it to predict process variables and be designed as if the process were free from delays. The Smith predicto**r** offers an improvement in closed-loop performance compared to regular controllers and has been widely applied to multiparametric systems with delays. However, like other model-based control systems, the Smith predictor requires an extremely accurate process model. When there is a discrepancy between the model and the actual process, the closed-loop performance can be severely compromised. In fact, studies have shown that Smith systems can become unstable even with infinitesimal disturbances in the “universal counts” used to select the closed-loop bandwidth, considering the uncertainty associated with process dead time. Nevertheless, these investigations did not account for the simultaneous uncertainties in gain coefficients, time constants, and dead time, which impact the reliability of Smith predictor controllers

A Smith predictor is a regulator capable of predicting the output signal. To do this, he can use the object model, which in turn consists of M0 and the delay e^{-sL} (Figure 2.1). Thanks to this, it is possible to eliminate the delay and it is possible to understand the behavior of the object before the output signal appears. This fact suggests that implementation is possible by the system presented in Figure 13, where R is the PID controller, P_0e^{-sL} is the transfer characteristic of the object.

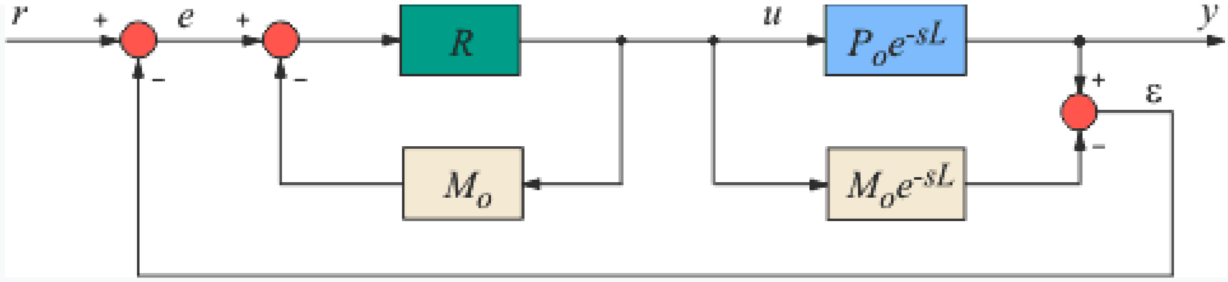


Figure 2.1 – System with Smith predictor

Based on the fact that we assume the model is accurate, the difference between the signals at the outputs of the model and the object is zero, ε=0, then from Figure 13 we obtain:

$$y = P_0 e^{-sL}\left(\frac{R}{1+RM_0}\right) r = \left(\frac{P_0 R}{1+P_0 R} e^{-sL}\right) \quad (2.1)$$

Here $\frac{P_0 R}{1+P_0 R}$ represents a function without transport delay. This indicates that the element with delay is absent in the control loop, and it does not impact the stability and performance of the system, as regulation occurs in the loop without delay.

Let’s now consider the operation of a predictor without the case where the difference is zero. This can be described by the following system of equations:

$$y = P_0 e^{-sL} R(r - \varepsilon - M_0 u), \quad \varepsilon = y - M_0 e^{-sL} u, \quad y = P_0 e^{-sL} u \quad (2.2)$$

From this, it follows:

$$y = \left[\frac{P_0 R}{1+RM_0+R(P_0-M_0)e^{-sL}}\right] e^{-sL} \quad (2.2)$$

Obviously, as the difference P0-M0 increases, the denominator tends to zero and the delay element is excluded from the transfer function, which is only added to the result.

Let's take a closer look at the output of the Smith regulator transfer function.

If you look at the part of the block diagram, which is designated as W in Figure 2.2, you can see that this part of the diagram consists of the fractional-rational part *M*0 and the PID controller R, let’s find the transfer function W for it:

$$e_1 = e - u_1, \qquad u = Re_1, \qquad u_1 = uM_0, \quad \frac{u}{R} = e - uM_0, \qquad u\left(\frac{1}{R} + M_0\right) = e,$$

$$W = \frac{u}{\varepsilon} = \frac{R}{1+RM_0} \quad (2.3)$$

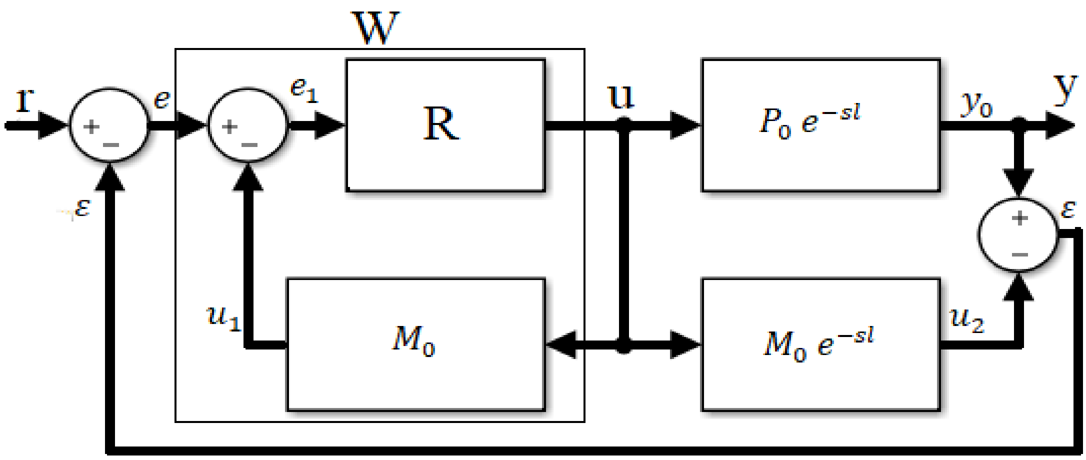


Figure 2.2 - Block diagram of a Smith regulator.

After identifying W, let’s modify the block diagram for convenience:

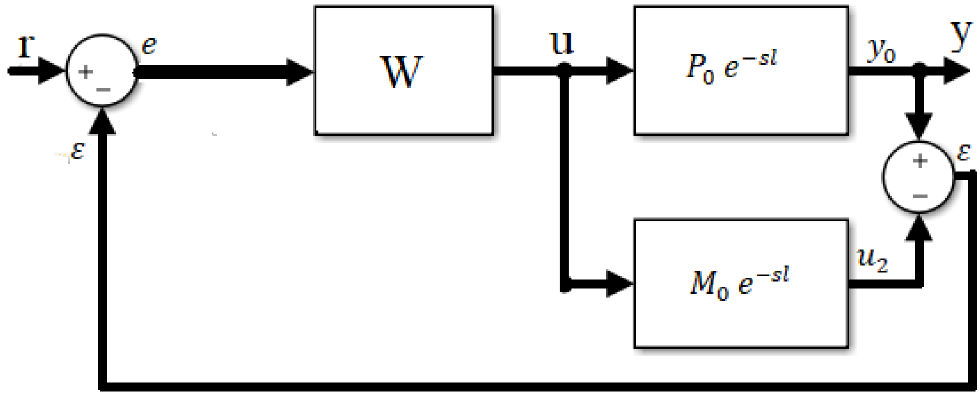


Figure 2.3 - Smith regulator structure after simplification

Next, we will find *W*total, which consists of the previously found W, the transfer characteristic of the control object with a delay $P_0 \cdot e^{-sl}$and the transfer characteristic of the object model with a delay *M*0 $M_0 \cdot e^{-sl}$.

$y_0 = u \cdot P_0 e^{-sl}$,

$u_2 = u \cdot M_0 e^{-sl}$,

$\varepsilon = y_0 – u_2 = u \cdot P_0 \cdot e^{-sl} – u \cdot M_0 \cdot e^{-sl} = u \cdot (P_0 \cdot e^{-sl} – M_0 \cdot e^{-sl})$,

$e = r – \varepsilon$,

$y = y0 = u \cdot P_0 \cdot e^{-sl}$,

$y = u \cdot P_0 \cdot e^{-sl}$,

$u = W \cdot (r – \varepsilon)$,

$\varepsilon = u \cdot (P_0 \cdot e^{-sl} – M_0 \cdot e^{-sl})$,

$u = W \cdot [r – u \cdot (P_0 \cdot e^{-sl} – M_0 \cdot e^{-sl})]$,

$u = W \cdot r – W \cdot u \cdot (P_0 \cdot e^{-sl} M_0 \cdot e^{-sl})$,

$$u[1 - W(P_0 e^{-sl} - M_0 e^{-sl})] = \frac{Wr}{1 - W(P_0 e^{-sl} - M_0 e^{-sl})},$$

$$u = \frac{Wr}{1 - W(P_0 e^{-sl} - M_0 e^{-sl})},$$

$$W_{общ} = \frac{W P_0 e^{-sl}}{1 - W(P_0 e^{-sl} - M_0 e^{-sl})} \quad (2.3)$$

Where$w = \frac{R}{1 + RM_0}$, R - PID Controller Transfer Function

It should be noted that at τ/T >10 values, this method of increasing the stability of PID control, with all its advantages, ceases to give stable control results.

2.2 Dynamic PID Controller Correction

The ideas of predicting the results of regulation and subsequent correction of the magnitude and time sequence of the influence of the regulator on the control object, which has large inertia or the value of the transport delay, similar to the Smith predictor, currently appear quite often. This work is also no exception. Accurate prediction of the result allows for dynamic correction of the PID controller parameters and thereby improves the quality of regulation by eliminating the causes of instability and optimizing the shape of the transient process.

One such way to optimize regulation is a genetic algorithm [12].

Genetic algorithms are a powerful optimization technique that can find the global optimal solution faster than other random search methods. Their main advantage is the absence of problems with stability and convergence. These methods are used to search for optimal positions of membership functions in phase regulators, to search for optimal controller parameters, to identify models of control objects, and for “training” in a neural network. In most cases, genetic algorithms are used in conjunction with fuzzy logic controllers and neural networks.

A disadvantage of genetic algorithms is that searching for extreme values takes a significant amount of time, rendering them unsuitable for high-speed real-time systems

Genetic algorithms are based on the principles of Darwinian natural selection, formulated in 1859. The idea of genetic algorithms for solving mathematical problems was formulated by J. Holland in 1962. Genetic algorithms use the concepts of genes, reproduction, selection, chromosomes, mutation, crossbreeding. The basic idea of genetic algorithms is directly similar to the principle of natural selection, where the most adapted individuals survive.

The main elements of natural genetics used for the search procedure are:

Crossover is the basic genetic operator that exchanges genetic material between individuals. Simulates the process of crossing individuals. and used for the search procedure are:

The concept of reproduction varies across different algorithms. It is contingent upon how data is represented and processed. However, a fundamental requirement for mating is that offspring must have the ability to combine characteristics from both parents in some manner

The mutation operator, used for selection and crossing of new generations, is used to select populations from a local extremum and protect against premature convergence.

Every bit of every individual in the population has a chance of flipping. This probability is usually very small, less than 1%.

Multiple points on a chromosome can be selected for inversion, and the number can also be random. You can also invert several consecutive points at once. usually recommended probabilities for choosing a mutation, you can often find options 1/L or 1/N.

Stopping criteria, in general, this evolutionary process can continue indefinitely. The stopping criterion can be the convergence of a given number of generations or the convergence of populations

Convergence is a state of a population when all rows of the population are in some extreme region and are almost the same. In other words, the crossover practically does not change the population in any way, and the mutating individuals tend to die out because they are less adapted. Thus, population convergence means that optimal solutions have been reached.

Certainly! Here's the translation of the classic genetic algorithm steps:

1. **Definition of the Fitness Function**: Determine a fitness function (also known as an objective function) for the individuals in the population.
2. **Initial Population Selection**: Choose an initial population of individuals, typically represented by chromosomes of size N.
3. **Evaluate Fitness**: Compute the fitness value for each individual based on the defined fitness function.
4. **Reproduction (Crossover)**: Apply reproduction operators such as crossover (recombination) to create new offspring from selected parents.
5. **Mutation**: Introduce random changes (mutations) to the offspring's genetic material.
6. **Crossover**: Combine genetic material from parents to create new individuals.

7. **Termination Condition Check**: Check if the termination conditions (e.g., convergence, maximum generations) are met. If not, repeat from step 2.

For the algorithm to work, you need to set the probability of mutation, the probability of crossing, the lower and upper limits of changes in the desired parameters, the maximum number of generations and the population size.

To design a PID controller based on a genetic algorithm, an initial population of chromosomes is randomly generated. This population contains binary strings, where the string represents the proportional, integral and differential part (Kp, Ki, Kd).

To determine the coefficients of a PID controller, one can be chosen, for example, based on minimum overshoot**,** minimum rise time, or as

$$J = \frac{1}{\int_0^t t|e(t)|dt} \tag{2.4}$$

Where: e(t) represents the current value of the error between the output signal and the reference signal, and **t** denotes time

In Figure 2.4, the structure of the algorithm for finding optimal coefficients of a PID controller using **a** genetic algorithm is presented

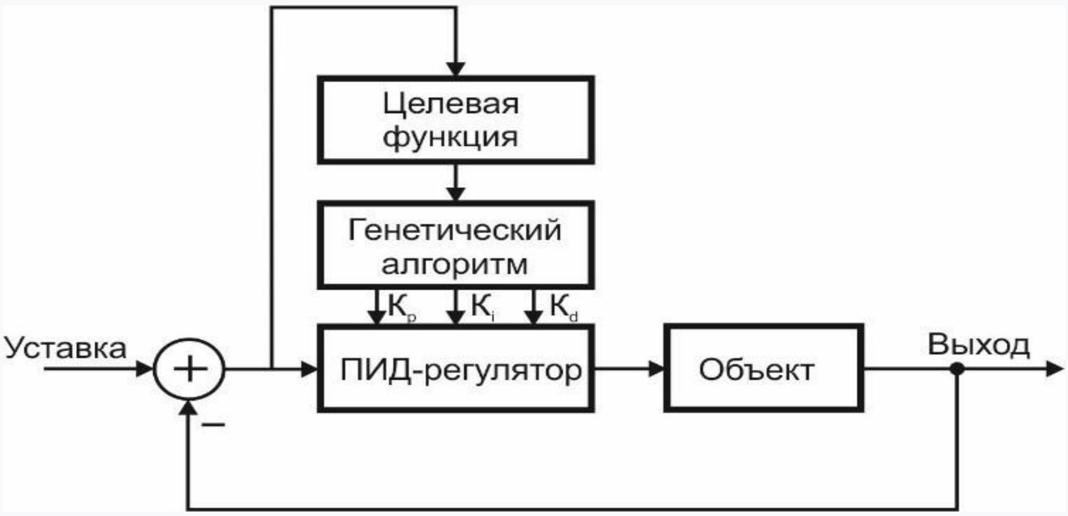


Figure 2.4 - Structure of a PID regulator based on a genetic algorithm

In the study, the search for optimal parameters of a PID controller was performed using a genetic algorithm with the following parameters: population size 100, crossover probability 0.9, maximum number of generations 100, and mutation probability 0.1.

The controller coefficients were obtained through modeling in MATLAB: Kp=1.3521; Ki=0.0174; Kd=0.07.

Figure 2.5 shows the results of the experiment on a real object with the obtained PID controller coefficients

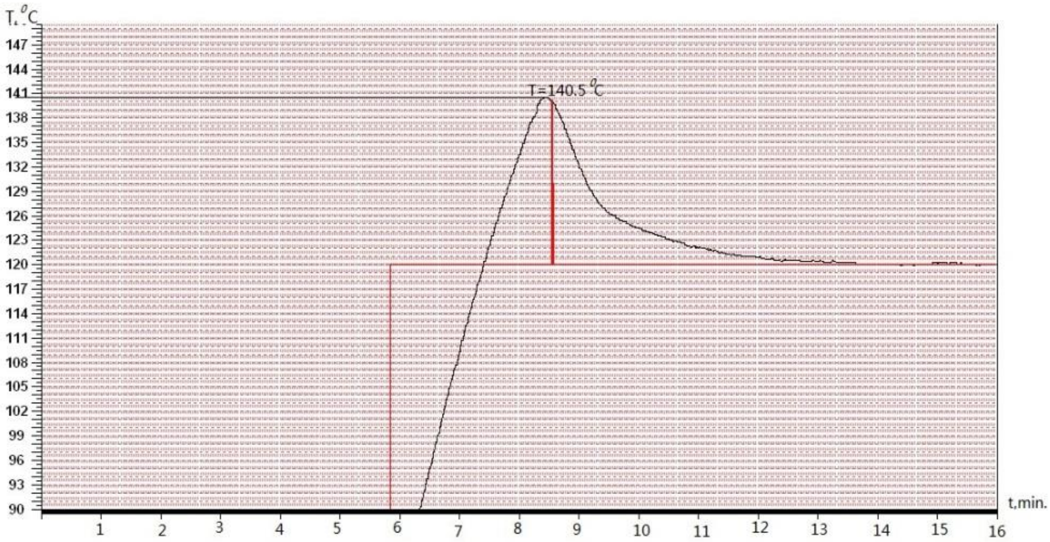


Figure 2.5 - Transient process in a system with a PID controller based on a genetic algorithm.

From Figure 2.5 it follows that an object using a PID controller, the coefficients of which are calculated based on a genetic algorithm, gives fairly good results, making the system stable. Overshoot is 16.7%. The regulation time is 400 seconds.

2.3 Three-position control of heating systems

Attention should be paid to the method of heating system regulation currently in use.

Heating systems have a very long transport delay time [13, 14], which is usually 10-30 minutes.

Industrially produced heating regulators do not allow the introduction of complex control algorithms, being limited to certain parameters: exposure time and cycle time. In this case, the value of the dead zone is usually 0.5-1.0 oC, determined by the sensitivity of the sensor used and is not standardized.

The block diagram of the controller is standard and is based on the principle of recording the deviation of the controlled variable ε from the specified value.

The algorithm for developing a control action on the regulator for supplying direct network water to the heating system of a building is constructed using a three-position method that does not have a dead zone using the signum function (2.5):

$$T_{обр}(t-\tau) = F[S - k(t)sign(\varepsilon)] \qquad (2.5)$$

τ - time delay

S - initial position of the regulating valve stem,

$\varepsilon = T_{зад} - T_{изм}$ - deviation of the measured temperature of the return network water from the set value, where the set temperature can vary depending on the outside air temperature

k(t) is a function of changing the position of the valve stem, which significantly depends on the specified exposure time and the specified cycle time between exposures.

The return network temperature target is presented as a graph of a function of the outside air temperature and is also entered into the heating controller.

The function of changing the position of the valve stem in one cycle can be represented as dependence (2.6.):

$$k(t) = \frac{z\omega\tau_{воздействия}}{\tau_{цикла}} \qquad (2.6)$$

here:

z - coefficient of the valve reducer,

ω - rotational speed of the valve stem actuator,

τ_actuation - specified time for the pulse delivery of voltage to the stem motor

The term “τcycle” refers to the specified time interval between two pulses

It should be noted that a significant transport delay time does not allow the use of a PID control algorithm or its modifications due to the resulting instability of regulation, leading to the regulator reaching an extreme state: either full closure or full opening of the throttle valve. Therefore, the valve opens with small pulses and high duty cycle or remains constantly open or closed depending on the sign of the signum function.

In Figure 2.6 a typical graph of the temperature variation of the return network water is shown for this control method

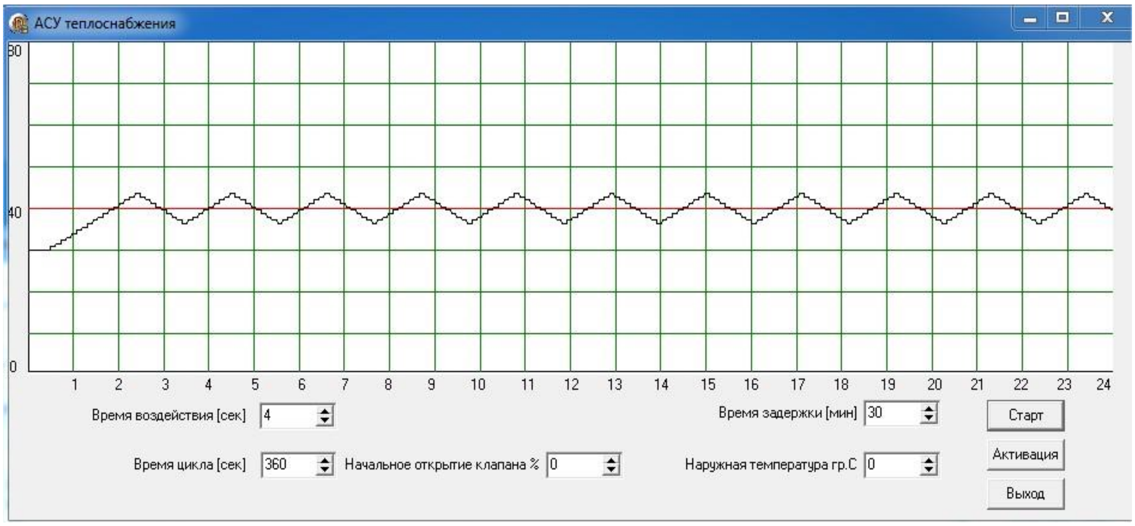


Figure 2.6 - Graph of changes in return network water temperature

It is obvious that the use of standard (recommended) settings of the heating regulator in terms of exposure time and cycle time does not provide the required quality of regulation, because At the same time, there are significant hourly fluctuations in the temperature of the return network water within 10 degrees, which cannot but affect the level of comfort in residential premises

In addition, it is important to consider increased wear and tear of the mechanical components of the regulator when operating under more demanding conditions compared to the optimal control mode.

In the presented work, an attempt is made to optimize such a regulator.

CHAPTER 3
CONSTRUCTION OF A PERCEPTRON CIRCUIT FOR REGULATING A HEATING SYSTEM

3.1. Artificial neural networks.

The purpose of this work is to increase the reliability of forecast estimates of the state of an object with a delay (in this case, the heating system of a building) based on neural technologies in order to develop the most optimal regulatory actions and provide the ability to train the network in order to reduce time and increase the reliability of the forecast.

Artificial neural networks (ANN) are one of the most successful tools for analyzing time processes and time series (TP).

The theoretical basis for the use of neural network models in the field of time series analysis is the fundamental theorem of Takens [15], which, based on the principle of repeatability of observations, made it possible to give an affirmative answer about the fundamental possibility of predicting future VR values based on a finite number of its previous values.

Takens' theorem, proposed by Floris Takens in 1981, provides a powerful framework for reconstructing nonlinear dynamical systems from a sequence of observations of their state. Here are the key points:

$$x(t+\tau) = \gamma[x(t), x(t-1), x(t-2), \dots, x(t-l), \vartheta_1, \vartheta_2, \dots, \vartheta_k] \qquad (3.1)$$

Where τ is the forecast lag;

l is the width of the immersion window;

k – the number of independent variables that determine the value of x.

Formula (1) allows predicting vague reasoning VR to be reduced to a typical problem of neuroanalysis - approximating a function of several variables for a given set of examples by immersing the series in a multidimensional space.

Figure 3.1 shows a general diagram of a neural network that implements the immersion method, called a time delay neural network (TDNN).

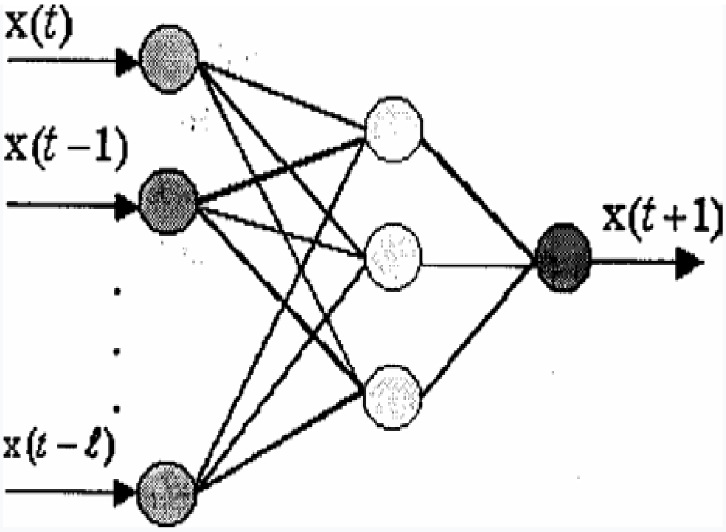


Figure 3.1 - Feedforward neural network with a time window as a nonlinear autoregressive model.

significant advancements in neural networks (NNs), specifically focusing on **recurrent neural networks (RNNs).** Unlike conventional NNs, RNNs incorporate **feedback loops,** allowing them to model **temporal data** by considering the context of their own functioning. This context includes previous output values generated by the network, which are then used as part of subsequent input to the network.

Typical representatives of recurrent ANNs are the Jordan-a network, shown in Figure 3.2,a, and the Elman-a network, shown in Figure 3.2,b.

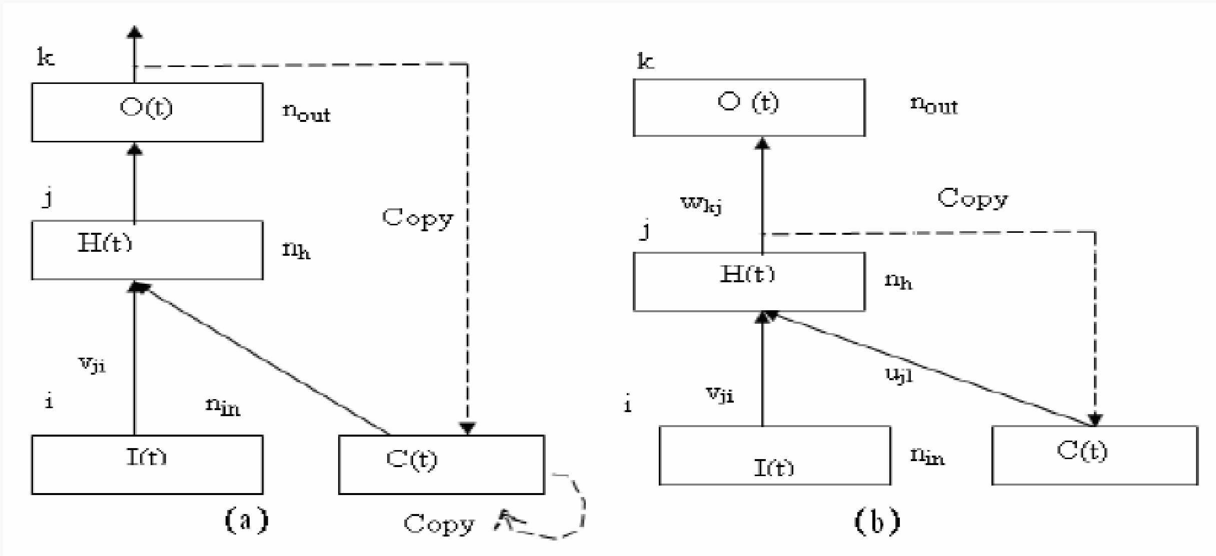


Figure 3.2 - Jordan network (a) and Elman network (b)

Recurrent neural networks (RNNs)**,** like ARIMA models, have their advantages and limitations when usedfor **time seriesforecasting.**problems, have a number of disadvantages associated with training, namely, they require a long training time and have an increased tendency to freeze.

Recurrent Neural Networks (RNNs), including a specialized type proposed in [16] known as **Multicontext Recurrent Neural Networks(MKRNs),** largely mitigate this limitation.

The architecture of the**Multilayer Contextual Recurrent Neural Network** (MCRS) is shown in Figure 21. It consists of four layers: input, hidden, output, and multi-context. The proposed architecture, combining the characteristic features of the Elman-a and Jordan-a networks, nevertheless has a distinctive feature - its multi-context layers are connected directly to the output layer, which reduces the dependence of the network output on hidden layers and speeds up the learning process.

Fuzzy Logic Systems (FLS), as a means of modeling **vague reasoning(VR),** represent, to some extent, an alternative to neural network models. FLS, like **Artificial Neural Networks (ANNs)**, are universal nonlinear data approximators and can be used as a tool for implementing a nonlinear autoregressive model that describes the dynamics of the investigated process. However, unlike ANNs, which operate based on the “black box” principle, FLS rely on easily interpretable linguistic or mathematically algorithmic information presented in the form of crisp rules. This allows for effective integration of prior knowledge into the model of the investigated process. This quality becomes extremely important when dealing with loosely structured processes, where information deficits can be partially compensated by a priori expert knowledge represented in the form of fuzzy rules within FLS

For studying dynamic processes, a special class of nonlinear systems(NLS) is used, which allows describing the process dynamics. Similar to recurrent neural networks (RNNs), the construction of dynamic NLS is based on the idea of transforming process dynamics into statics using an immersion method. For this purpose, a regular NLS is supplemented with feedback inputs to which time-delayed signals are applied. The resulting fuzzy system implements a nonlinear mapping from preceding values of the time series to subsequent ones.

The implementation of expression (3.2) in NLS is based on the well-known formalism of fuzzy logic, which represents a general framework for modeling clear or fuzzy information, the core of which is fuzzy rules of the form “IF <<CONDITION>>, THEN <<Action>>”. The preconditions and conclusions of the rules contain, respectively, the preceding and subsequent BP values, presented in the form of linguistic terms.

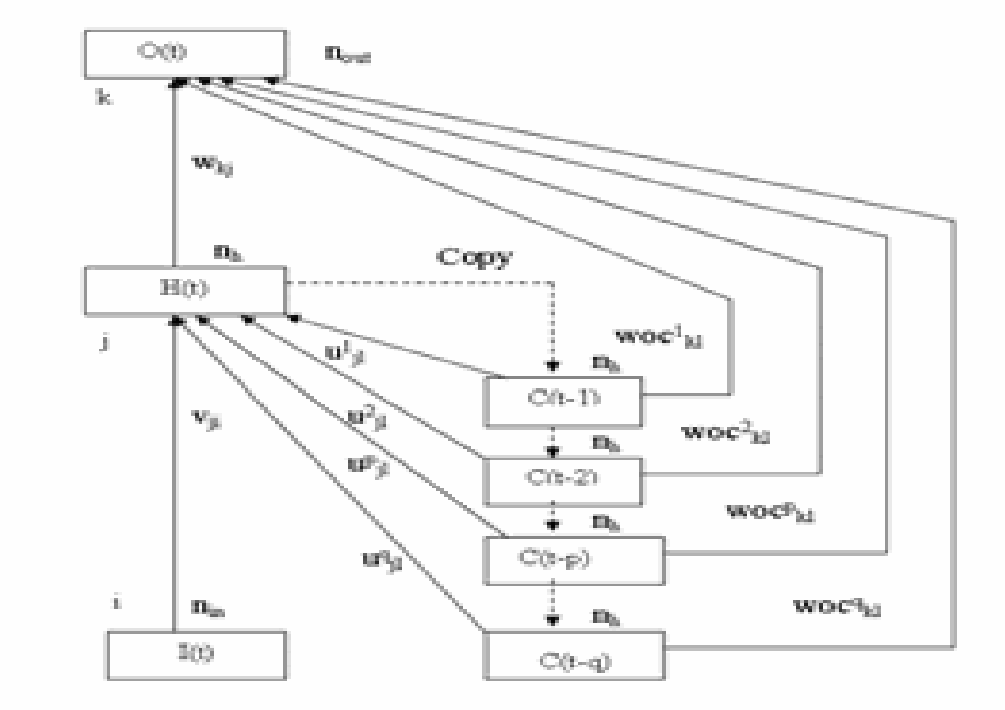


Figure 3.3 - Architecture of a multi-context recurrent neural network

$$X(t-1) \times X(t-2) \times \dots \times X(t-k) \rightarrow X(t+1) \quad (3.2)$$

The functioning of the natural language systemsNLS comes down to the implementation of particular implications for each of the rules, followed by combining the results to obtain the final solution. The resulting fuzzy solution can be converted by defuzzification into a specific predicted value of the series.

The defuzzification module, which is typically an essential component of the overall structure of any fuzzy logic system (FLS)**,** is not mandatory for fuzzy systems implementing nonlinear autoregression processes. This is because, for tasks related to time series forecasting, there is no need to convert the results to a crisp value. Furthermore, for the purpose of enhancing system stability and reliability, it proves useful to retain intermediate fuzzy inference results in the same form as obtained in preceding iterations and utilize them as input for forecasting subsequent iterations.

The fuzzy inference, implemented using input data represented not as specific numerical values but as fuzzy subsets of such values, is referred to as non-singleton fuzzy inference**.**

NLS with non-singlet output, being a generalization of traditional NLS, provide an easily mathematically processed tool for working with input uncertainty, both during the direct functioning of fuzzy systems and during their training. Having been proposed for the first time in [15], NLSs with non-singleten output are now increasingly being used in a wide variety of applications.

An example of the network structure of a dynamic NLS with a non-singlet output is shown below in Figure 3.4.

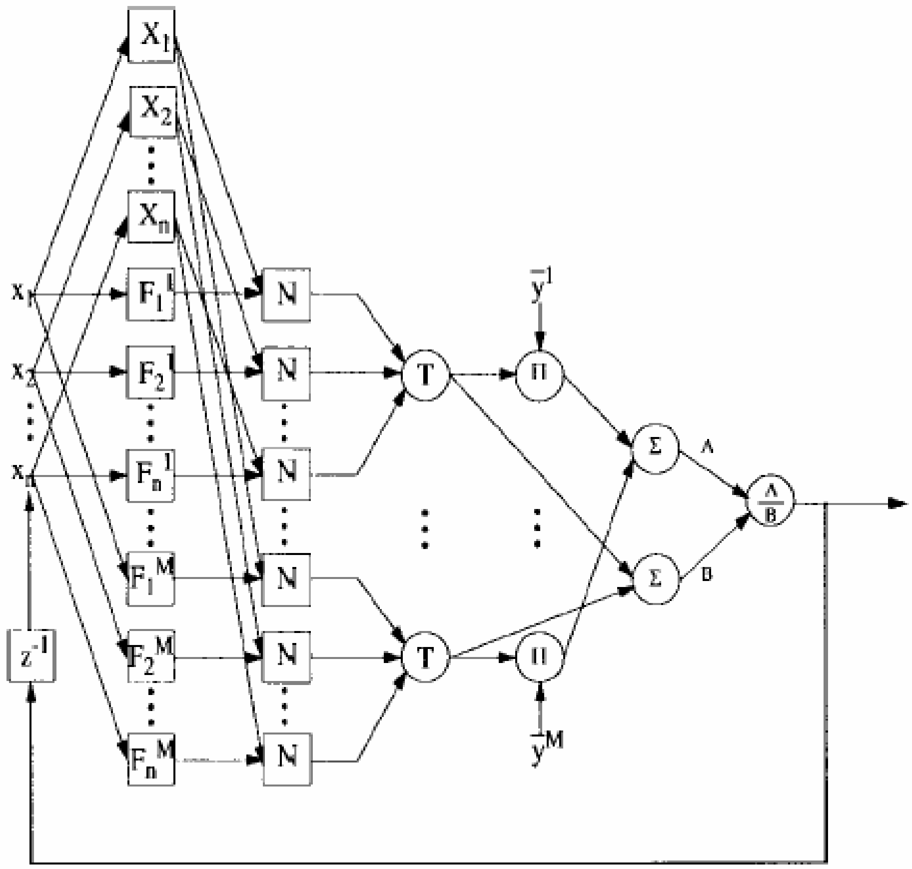


Figure.3.4 - Network structure of a dynamic NLS with non-singlet output with M rules (n-1) external inputs and one feedback input X_n.

Experimental studies with dynamic NLS (Nonlinear Systems) featuring non-singlet output have demonstrated clear advantages over their corresponding static fuzzy systems in modeling processes of unknown order and structure. Much of this advantage stems from the inclusion of additional feedback input, which allows for a more accurate reflection of process dynamics when its structure is unknown. Additionally, attempts have been made to view dynamic NLS in the context of belonging to the general family of nonlinear autoregressive moving average models. The results indicate that when forecasting time series, dynamic NLS, operating jointly with the components of an autoregressive model alongside moving averages, exhibit lower mean squared prediction error compared to their static fuzzy counterparts.

3.2 Construction of a perceptron for a heating system.

The application of traditional multidimensional nonlinear models that implement genetic algorithms and require multiple and multivariate calculations often proves to be inefficient. Therefore, in this study, an approach to modeling complex network structures of dynamic NLS (Nonlinear Systems) with non-singlet output was employed, similar to the one discussed in section 3.1.

A new approach to modeling complex systems is based on artificial neural networks (ANN). It allows you to find a solution in the absence of a priori knowledge about the laws of the modeled process. The main ideas underlying neurocomputing and neuromodeling are the following:

The concept of artificial neural networks **(ANNs)** involves replicating the structure of the nervous system in living organisms. A large number of simple computational elements (neurons) form a structure (the ANN) that exhibits more complex behavior and capabilities compared to those of individual network neurons.

An artificial neural network, just like a natural biological one, can learn. To do this, the ANN must be adaptive and change its resulting behavior.

A neural network learns to solve a new task using a set of situations (a dataset), where each situation describes input signals to the network and the corresponding output response. This dataset consists of problems with known solutions, and during training, the neural network is equipped with a knowledge base that establishes the relationship between input signals and responses

The structure of the ANN (its architecture) can be adapted to the problem being solved. Additional neurons can be included in the ANN if the initial network is not capable of solving the problem with the required accuracy.Extra neurons and connections between them can be excluded from the ANN if the initial structure is redundant for solving a given problem - this can simplify the subsequent electronic implementation of the ANN and increase the speed of its operation. The ANN can itself select the most informative input signals for the task, allowing uninformative, noise signals to be discarded, thus reducing the cost of collecting information and increasing the stability and reliability of the solution.

ANN models have a number of advantages that allow them to be widely used in analysis and forecasting problems, namely:

• no need to build a mathematical model of the analyzed process (formation of equations of connection between variables);

• the ability to restore nonlinear functional dependencies between the studied parameters (characteristics);

• effective work in conditions of incompleteness of initial information;

• the possibility of using small training samples that do not allow obtaining statistically reliable results based on classical methods;

• quick response of a trained neural network to the receipt of current information (at the level of processing telemetry data using existing methods);

• ensuring an almost full operating range;

• taking into account an unlimited number of influencing factors;

• no need for equivalence;

• high degree of capacity of the model;

An artificial neural network is a set of neurons connected to each other using a specific architecture called a perceptron.

The input layer of the perceptron will represent a set of coolant temperature values in the return line.

The hidden layer, the intermediate layer, determines the weight of the connection between the neurons of the hidden layer and the neurons of the output layer. This layer will have a set of outside air temperature values, according to which a connection will be established with the neurons of the output layer.

The output layer is the most significant. Here will be an information base of impacts on the regulatory elements of the heating network:

• electric power supplied to the heating elements;

• the opening value of the valve in the direct heating network.

The actual values of the regulatory influences are determined experimentally during the so-called "training" period of the perceptron model. The use of these values to implement control should bring the heating system to a given state in accordance with the approved heating schedules without additional regulatory influences.

We construct a perceptron in the form of a radial basis function. Its diagram is shown in Figure 3.5.

The perceptron is created to bring the heating system from the initial state to a stationary mode with the least energy loss. It can also be activated when there is a sudden change in outside temperature.

In program form in the C programming language, the perceptron circuit is

implemented in the form of nested procedures like if elseif{if .. elseif{} .. }etc.

The so-called "training" of the perceptron is performed by adjusting the impact database by 3-5% when there is a significant (over 5%)in case of a significant (over 5%) deviation of the controlled temperature from the set values after three control cycles, followed by equalization of the values throughout the entire layer.

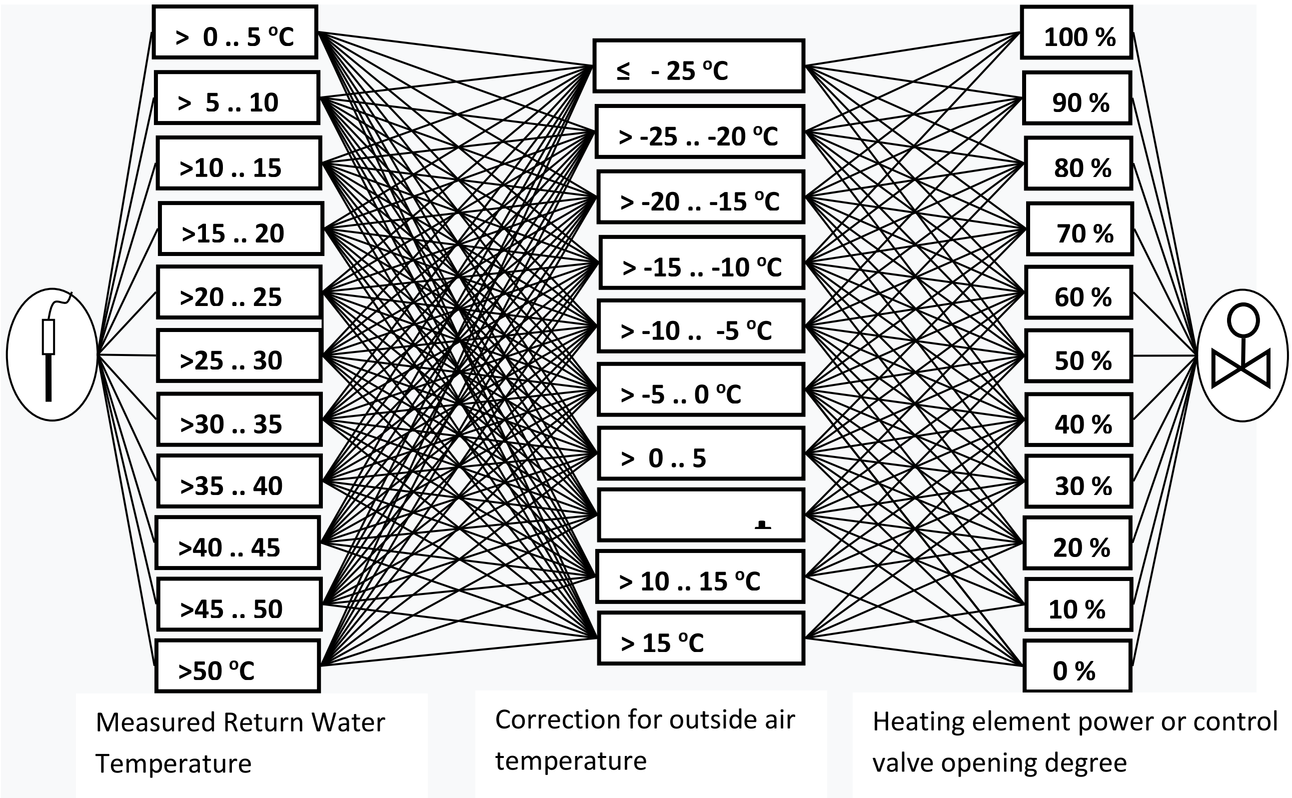


Figure 3.5 - Block diagram of the radial basis function of the system heating regulation.

The frequency of measuring the temperature of the return network water for regulation should be greater than the transport delay time. Otherwise, the regulatory process will be unstable.

For this reason, there is no point in studying the first and second derivatives of the temperature deviation from a given one: they may depend on a large number of random factors, which cannot be taken into account.

The parameter fuzzification scheme is presented in Figure 3.6.

Triangular and S-shaped membership functions are used for fuzzification.

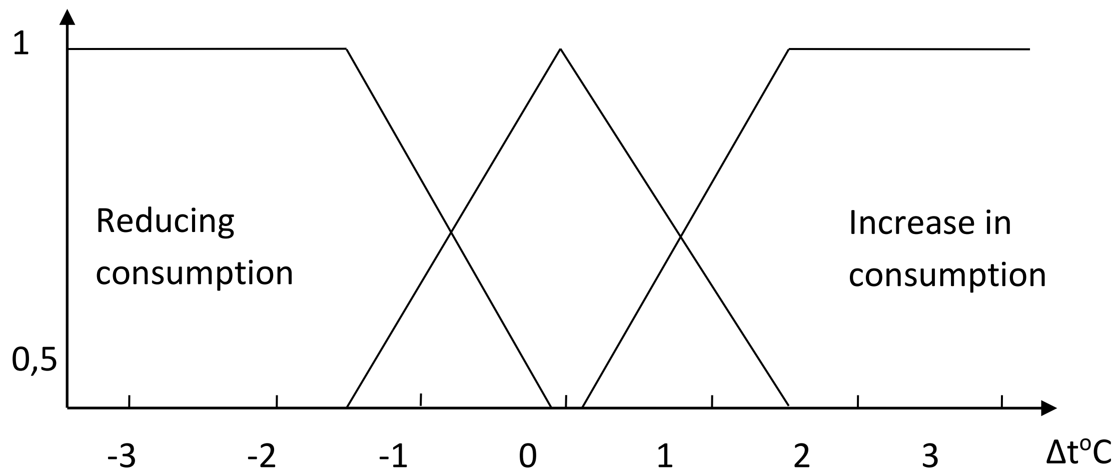


Figure 3.6 -. Graph of the membership function of the control system.

Defuzzification is performed using the center of mass of the graph figures [4,6].

The magnitude of the impact on the regulator in each control cycle is determined by formula (3.3.).

$$F = -SL\frac{\sum \mu(x)\Delta t}{\sum \mu(x)} \tag{3.3}$$

In this formula, the value of full opening of the valve L is determined by the response time of the regulator from full closing to full opening.

The "-" sign determines the negative feedback of the regulator.

The value S is the step action on the controller in each control cycle. When the controller starts, it is equal to 0.01. In subsequent control cycles, this value is refined using a specially developed algorithm (for example: it decreases or increases every 10 cycles depending on the Δt values).

3.3 Development of a perceptron training scheme.

The basis for the perceptron training circuit is a dynamic NLS circuit with non-singlet output with M rules (n-1) external inputs and one feedback input, shown in Figure 3.4.

The following are used as external connections for the heating system:

- measured values of ambient air temperature,
- measured values of temperature of direct network water.
- controle valve opening value or flow rate value.

Direct network water for heating. However, existing valves do not have a calibrated opening value scale and no remote signal generation associated with it. Therefore, we will determine the opening value by the engine operating time, which must be entered into the corresponding variable. The current value of this variable must be calibrated regularly, since the accuracy of its assessment is low.

Rules for determining internal connections of perceptron M:

- graph of changes in the temperature of direct network water depending on the outside air temperature (shown in Figure 25),
- graph of changes in return network water temperature depending on the outside air temperature (also shown in Figure 25),
- calculation of the basic value of the controle valve opening depending on the outside air temperature F=kt, where the value of k is determined experimentally,
- determination of additional corrections ΔF to the valve opening value is determined by formula (3.3) during the operation of the perceptron and is entered

into the knowledge base and is subsequently used to correct the basic valve opening value F+ΔF.

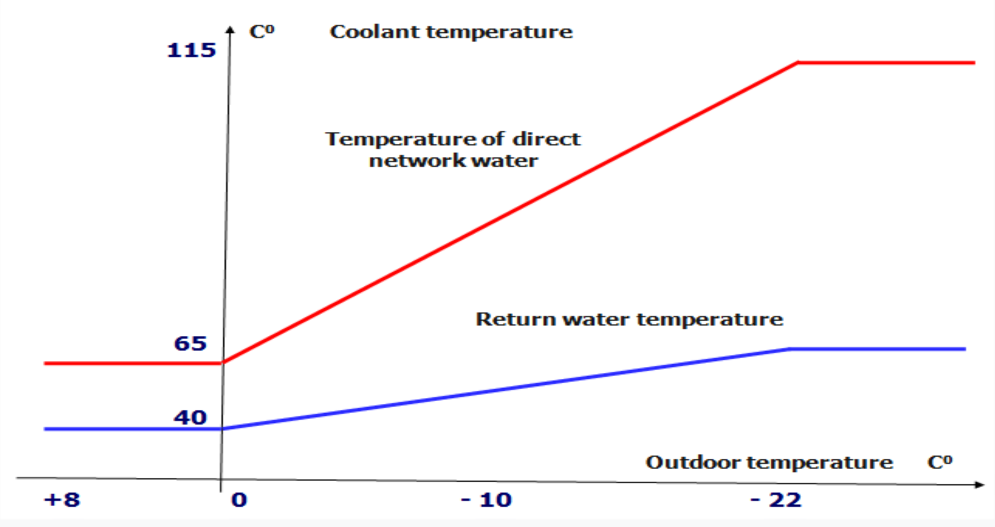


Figure 3.7 - Temperature graphs of the building heating system

The feedback input is represented by the value of the return network water temperature, which must correspond to the calculated value of the setpoint.

The perceptron connection diagram, presented in the form of a basic control function, is shown in Figure 3.7. This diagram shows the results of using the rules as separate layers

The general scheme for implementing fuzzy control and perceptron training is presented in Figure 3.8.

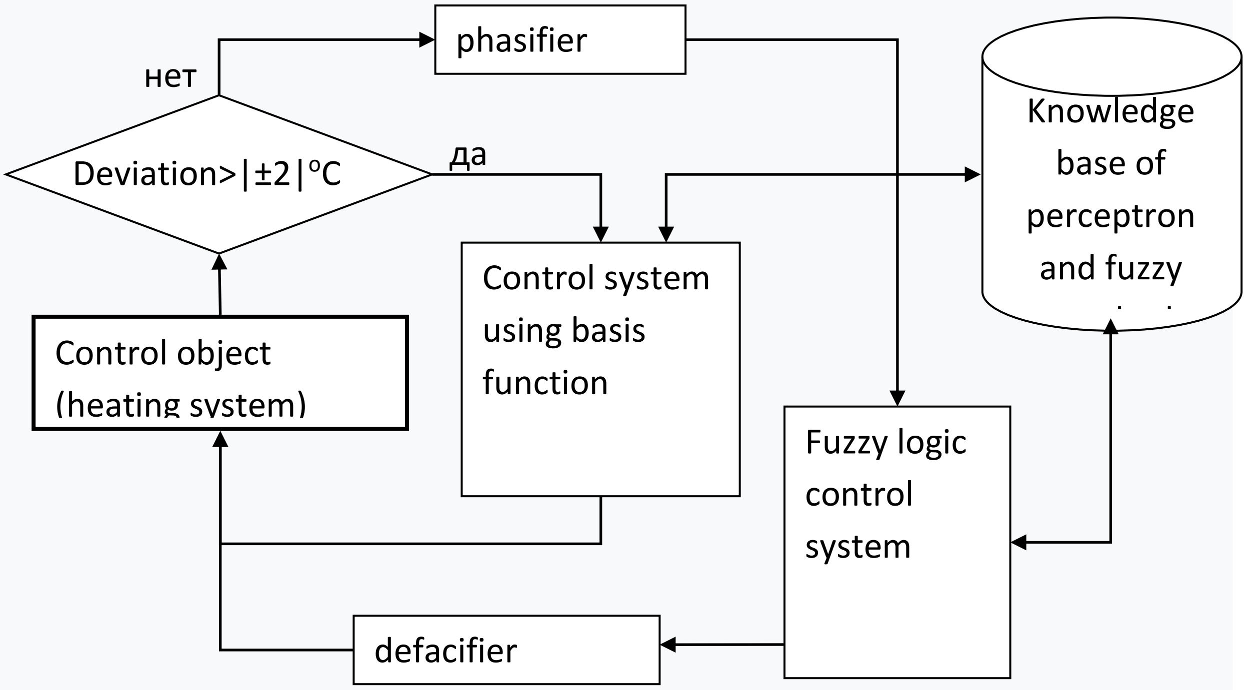


Figure 3.8. Diagram of a heating control system with fuzzy control.

The perceptron knowledge base is made in the form of a 3-dimensional matrix df[i,j,k] and contains the results of corrections ΔF to the base value, respectively, for a set of 3 measured temperatures: i - temperature of forward network water, j - temperature of return network water, k - outside air temperature. The meaning of variable numbers is determined by converting the variable format to the format of unsigned integer values.

It should be noted here that in addition to the internal connections taken into account, there are also unaccounted for ones, such as the influence of changes in wind blowing heat out of the building, or the influence of solar radiation and others. The magnitude of corrections in the knowledge base during the operation of the perceptron will change accordingly.

CHAPTER 4
EXPERIMENTAL PART OF THE WORK

4.1 Diagram of the stand for testing the operation of the FL-regulator algorithm

To implement the software part of the perceptron, using regular PLC controllers is not feasible, as the memory capacity and the available algorithm blocks do not allow for the creation of complex control algorithms with a large-dimensional mutable database. Therefore, the software component of the controller was developed for an AVR controller from Atmel (specifically, the ATmega 32-8P-AU, assembled on an Arduino NANO board)

To study the developed control algorithm, as well as other algorithms outlined in the work, it was most convenient to use the existing stand, which is a control object with a delay. The stand contains 2 containers: E1 - main container, E2 - auxiliary. The main tank contains a liquid level sensor (3) of a float type and a potentiometric circuit that provides an output voltage of 5-15 volts, proportional to the liquid level.

The control algorithms were tested on the “Water Supply” stand produced by NTP “Center” (Mogilev, Republic of Belarus), which is shown in Figure 4.1.

Figure 4.1 - General view of the stand, where it is indicated

1 – pumping unit, which includes a centrifugal pump (H) and a single-phase drive electric motor;

2 – pneumatic tank (PB);

3 – hydraulic control module (MGU);

4 – electrical control module;

5 – upper hydraulic tank (B2);

6 – potentiometric level sensor (DL);

7 – water level indicator in the upper hydraulic tank (UH);

8 – lower hydraulic tank (B1);

9 – water level indicator in the lower hydraulic tank (replaced with a tubular indicator);

10 – stand frame.

As a regulating element, a through-seat control valve of the VKSR Dn15 brand, manufactured by VogezEnergo LLC, was used. Schematically, the test bench can be represented as follows:

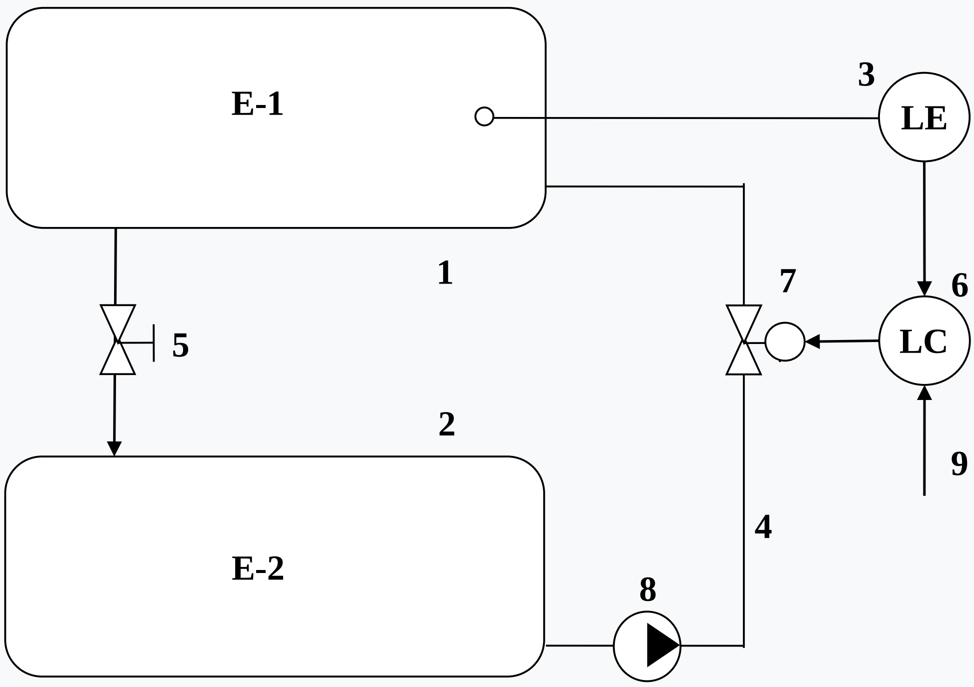


Figure 4.2 - Schematic representation of the control object

1 – main tank; 2 – drain container; 3 – liquid level sensor; 4 – delay line; 5 – full bore ball valve; 6 – level regulator; 7 – control valve drive; 8 – pump; 9 – setpoint

The pump (8) ensures that the liquid is poured through the pipeline (4) into the main tank from the auxiliary tank. The liquid is transferred to the drain container by gravity by opening the ball valve (5).The unfilled pipeline and part of the main tank up to the level of the float acts as a link with a pure delay. The formation of analog input signals of the level and level setter occurs as a result of the receipt of measurement results in the controller. This regulator controls the valve capacity by changing the flow through pipeline 4.For this object, the disturbing influences are the value of the hydraulic resistance of the valve (it is set by the regulator) and the opening value of the manual valve 5, which significantly affects the level in the main tank.

The purpose of regulation is to fill and ensure a given level in the upper (main) tank.

The Atmel AVR controller (ATmega 32-8P-AU) solves a number of typical problems:

- measurement of input analog signals;
- analog-to-digital signal conversion;
- computational operations using a PID algorithm or perceptron;
- generation of output discrete control signals.

To create a control program, the Arduino environment standard IEC 61131-3 is used, which contains a number of special drivers for communication with a PC. This environment is widely used in the field of automation and digital signal processing.

We will identify the automation object

Identification of a control object is the process of obtaining a mathematical model of the object. One of the widely used methods for identifying control objects is the graphical method based on the transient response or acceleration curve. Acceleration curve is the curve obtained as a result of the object's response to an input step effect.The acceleration curve or transient response h(t) is the response of the control object to a step input action. This acceleration curve is used to determine the dynamic characteristics of the control object.

Before identifying the system, it is necessary to drain all the liquid from the main one into the auxiliary one.

The process of obtaining the acceleration curve is as follows.

which is permanent. If we imagine that the link is ideally integrating (with an infinite amount of water and the volume of the container), but there are a number of restrictions (the amount of water and the volume of the container, and the limited possible volume of pumping by the pump).

The object can be represented as an ideal integrating link in the presence of an infinitely large capacity and an unlimited amount of liquid. To determine the acceleration curve graph, water is poured from the main tank into the auxiliary tank to a minimum. Then, the regulator is switched to manual mode, where it is necessary to record the time of supply of the step action, that is, the change in the controlsignal to the valve actuator, which can significantly increase or decrease the performance of line 4. Figure 28 shows a graph of the system acceleration curve. Input step action - manual opening of the valve to half the range of movement of the valve stem.

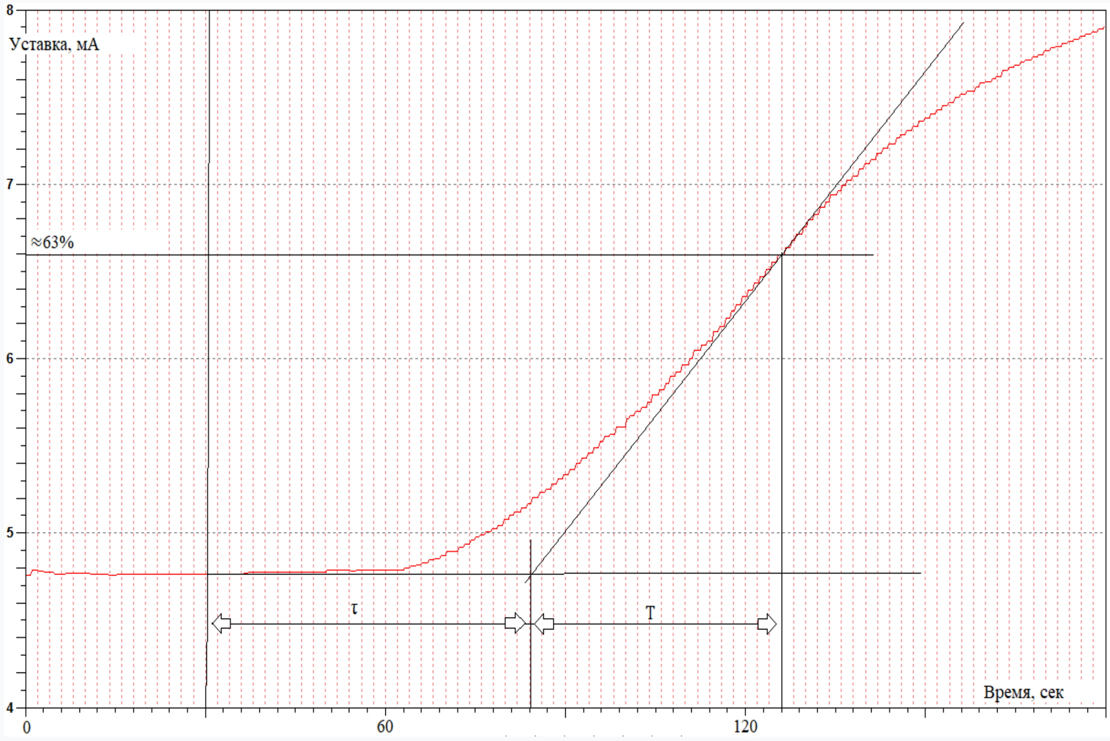


Figure 4.3 - Acceleration curve of an object under stepwise input action

The transfer function of the control object is taken in the form of an aperiodic link with pure delay:

$$W(s) = \frac{k}{Ts+1} e^{-\tau s} \tag{4.1}$$

where τ is the time interval from the moment the input signal is supplied until the start of changes in the system system (or delay time);

T – object time constant;

k – transmission coefficient (the ratio of the steady-state value of the output value to the value of the step input signal, k = 1.3).

From this curve, the delay time τ=55 seconds and the time constant T=47 s are determined.

Thus, taking into account the obtained values, we can write the transfer function of the control object in the following form:

$$W(s) = \frac{1,3}{47s+1} e^{-55s} \tag{4.2}$$

When setting up the controller using traditional methods of setting up the controller, difficulties may arise, since the ratio τ/T>1. This ratio is a guideline for the design engineer when choosing the type of regulator.

If $0.2<\tau/T<1$, it is recommended to use a PID or PI controller (continuous or digital), if $\tau/T<0.2$ then a digital or continuous controller.

If this ratio is greater than one, then there is a need to compensate for the delay in the control system loop.

It should be especially emphasized that the delay link of this stand is active only at the initial moment of time when starting the filling mode. Subsequently, when the tank is filled to the sensitivity level of the level control circuit, this link is excluded from the control circuit. Therefore, we use the object model with a delay link only in the initial period.

4.2 Description of the sketch program for the ATmega 32P controller.

Creating a program is not difficult. The text of the program is written in C. A special feature of the program is that the main module of the program assumes its constant execution in an endless loop, so it is marked not as <main>, but as <loop>. Therefore, to align operations in time, it is necessary to use a large number of delays or use various variants of timing functions.

Below is one version of such a program, written to implement the FL algorithm in its simplified interpretation without implementing the training option.

```
#define MOTOR_IN 2                                   // opens the valve
#define MOTOR_OU 4                                   //close the valve
unsigned long l_time;
int ih, is, i;
float h1, h2, dh, s;
void setup() {
pinMode (MOTOR_IN, OUTPUT);                          // gives s signal
pinMode (MOTOR_OU, OUTPUT);                          // gives s signal
digitalWrite (MOTOR_IN, HIGH);                       // initial setting - passive
digitalWrite (MOTOR_OU, HIGH);                       // initial setting - passive
Serial.begin (9600); // reading speed
i=0;
ih = analogRead(A4);                    // read the current level in the tank
is = analogRead(A6);                    // read the controller readings
h1=(ih-500.0)*2.0*50.0/1023.0;
h2=(ih-500.0)*2.0*50.0/1023.0;
}
// ************************************************
void loop() {
```

```
ih = analogRead(A4);                    // read the current level in the tank
 is = analogRead(A6);                   // read the controller readings
h2=(ih-500.0)*2.0*50.0/1023.0;
 s=50.0*is/1023.0;
Serial.print ("  Set ");
Serial.print (s);
Serial.print ("  in tan ");
Serial.print (h2);
Serial.println (" liters");
 dh = h2-h1; //dh>0
l_time = millis();
if(h2>s+10.0){                    // 1 - level is higher than the setpoint10
digitalWrite (MOTOR_OU, LOW);
delay(10000);                              // 10 seconds closes
digitalWrite (MOTOR_OU, HIGH);
i=1;
} else
 if(h2>s+3.0 && dh>0){ // 2 - level is higher than the setpoint 3 and growing
digitalWrite (MOTOR_OU, LOW);
delay(5000);                               // 5 seconds closes
digitalWrite (MOTOR_OU, HIGH);
i=2;
} else
 if(h2>s+1.0 && dh>0){  // 3 - level is higher than the setpoint 1and growing
digitalWrite (MOTOR_OU, LOW);
delay(2000);                               // 2 seconds closes
digitalWrite (MOTOR_OU, HIGH);
delay(3000);                              // 3 seconds shutter speed up to
5
i=3;
 } else
 if(h2>s+1.0 && dh<0){        // level is higher than the setpoint 1 and falls
delay(5000);                             // 5 seconds shutter speed up to 5
i=4;
 }
            // ----------------------------------------------------------------------

if(h2<s-1.0 && dh>0){     // the level is less than the setpoint by 1 and is
increasing
```

```
delay(5000);                              // 5 seconds shutter speed up to 5
i=5;
 } else
if(h2<s-1.0 && dh<0){       // the level is less than the setpoint by 1 and falls
digitalWrite (MOTOR_IN, LOW);
delay(2000);                                          // 2 seconds opens
digitalWrite (MOTOR_IN, HIGH);
delay(3000);                              // 3seconds shutter speed up to 5
i=6;
 } else
if(h2<s-3.0 && dh<0){       // the level is less than the setpoint by 3 and falls
digitalWrite (MOTOR_IN, LOW);
delay(5000);                                          // 5 seconds opens
digitalWrite (MOTOR_IN, HIGH);
i=7;
 } else
if(h2<s-10.0){                           // the level is much less than the setpoint
digitalWrite (MOTOR_IN, LOW);
delay(10000);                                         // 10 seconds opens
digitalWrite (MOTOR_IN, HIGH);
i=8;
 } else delay(5000);
 h1=h2;
Serial.print ("The mode was running ");
Serial.print (i);
}
```

The type of transient process obtained when entering this control program into the controller is presented in Figure 4.8.

4.3 Results of experimental implementation of control algorithms.

Since it was not possible to test the algorithms under study in natural conditions on a real heating system of a building, a stand was used as a testing tool, the description of which is given in Section 4.1, and in natural conditions the algorithm was checked manually on an operating thermal automation system

First of all, it makes sense to look at the operation of the stand with a software implementation of the PID controller algorithm installed in the controller and with the transport delay link excluded by drawing water into the main container to the average level, bypassing the delay link (Figure 4.4).

It can be seen that for such systems the PID control algorithm can achieve excellent results.

After this, with the same parameters of the PID algorithm, we introduce a transport delay. We start the regulation process with the main tank empty. The results are presented in Figure 4.5.

It is obvious that the PID algorithm cannot cope with the assigned functions: a large overshoot of the level is allowed, the system as a whole is on the border of stability.

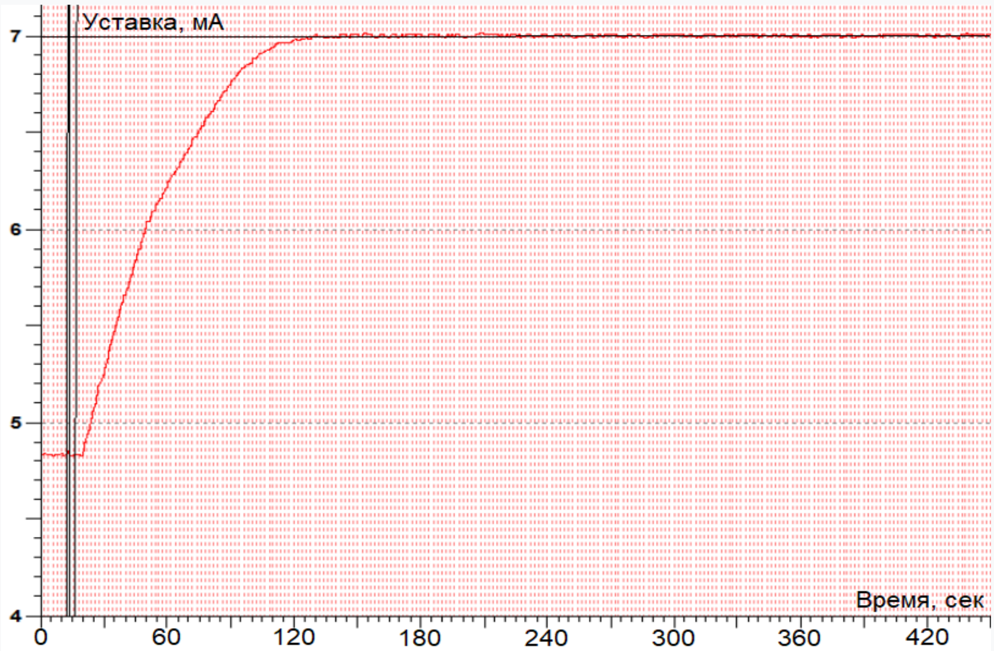


Figure 4.4 - Transient process of the stand with the lag element excluded, with PID controller coefficients (proportional - 50; integral - 25; differential - 1).

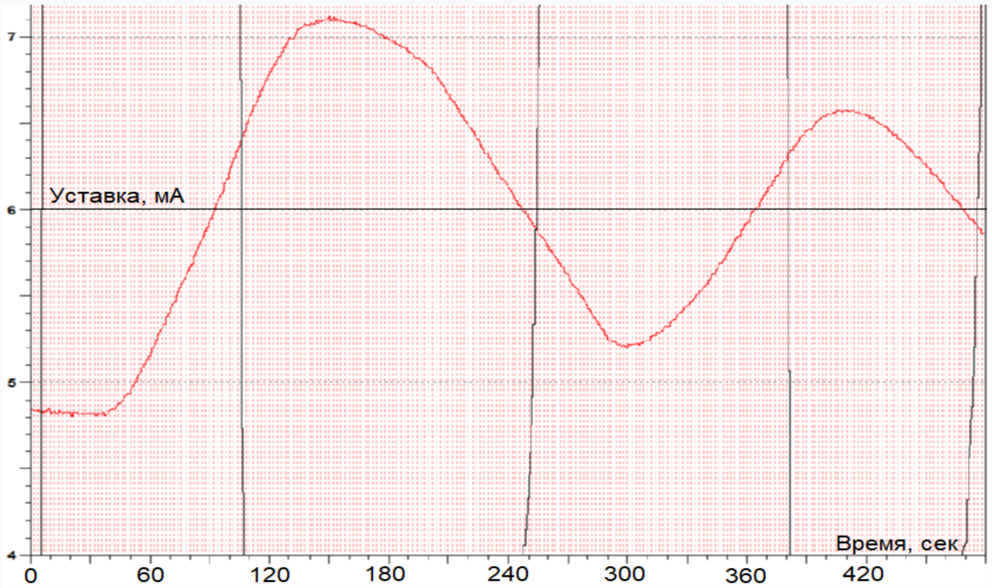


Figure 4.5 - Transient process of the stand with the delay link turned on, with PID controller coefficients (proportional - 50; integral - 25; differential - 1).

Let's move on to studying the regulation of a technological process with transport delay, where the Smith predictor is used as a regulator.

Let us implement the control algorithm with the Smith predictor described in Section 2.1 for an object with a delay.

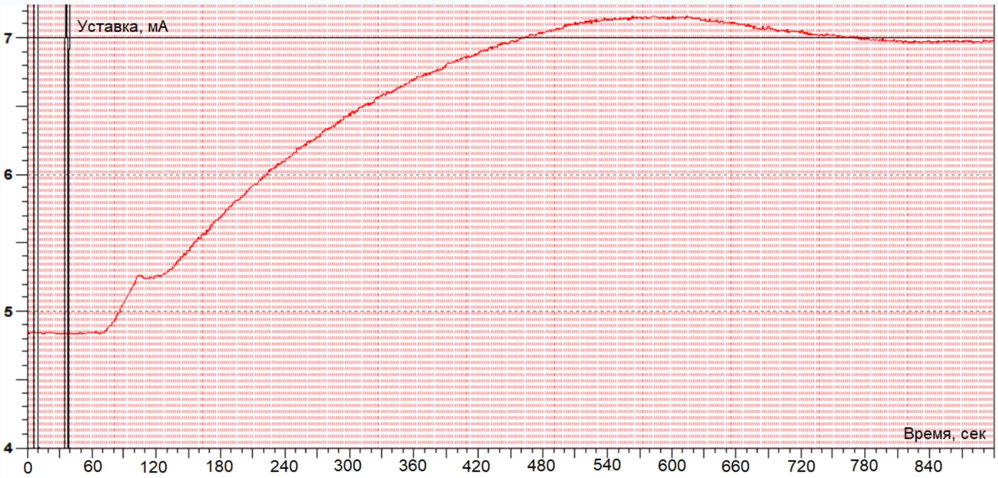


Figure 4.6 - Transient process of the stand with the delay link turned on, with a Smith regulator, PID controller settings (proportional - 50; integral - 25; differential - 1).

The transient process (Figure 31) with these coefficients can be characterized as monotonic, with a very short overshoot time for this system (~240 seconds) and a time to reach the set point of ~730 seconds.

Let's check the relay algorithm for controlling an object with a delay, described in Section 2.3. Let's set the exposure time to 30 seconds and the pause to 30 seconds. The results can be seen in Figure 4.7.

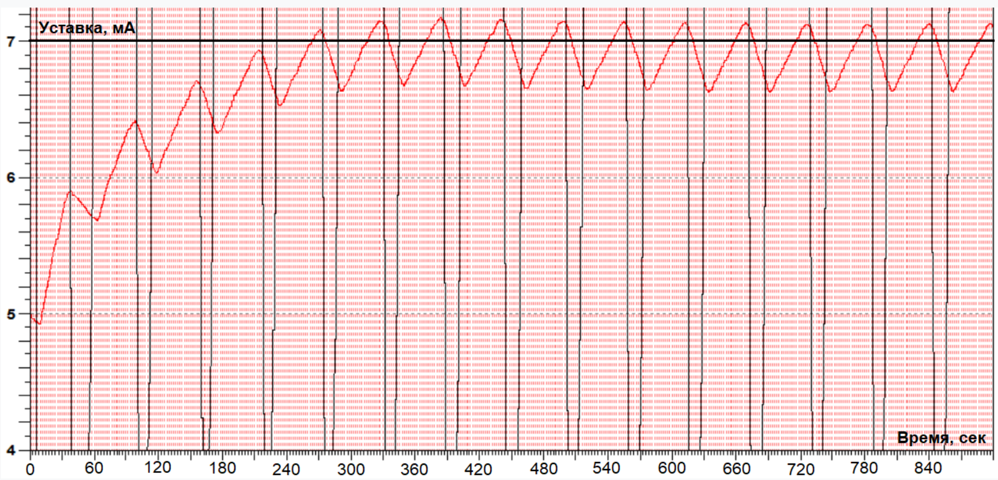


Figure 4.7 - Transient process of the stand with the delay link turned on, the relay control algorithm is configured for an exposure time of 30 seconds and a pause time of also 30 seconds.

It can be seen that the regulation process, while remaining stable and without overshoot, requires constant intervention from the regulator, creating a sawtooth process characteristic of relay systems. The magnitude of level fluctuations can reach significant values. Reducing the exposure time or increasing the pause, although it reduces the control error, leads to a significant increase in the control time (10 times or more). With further changes in parameters, it is not possible to achieve the specified level at all.

Figure 4.8 shows the results of an experimental test of regulation using a perceptron, implemented as presented in Section 4.2. programs.

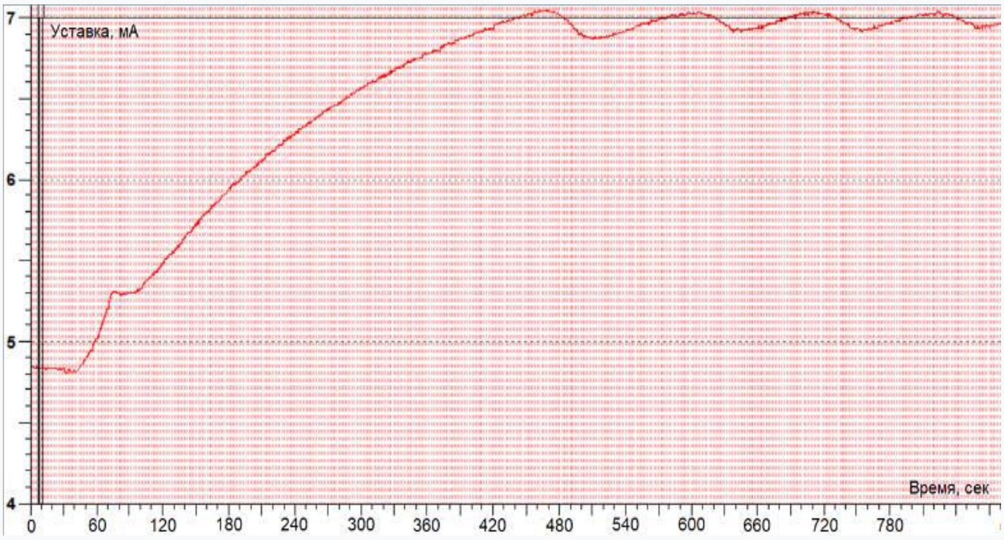


Figure 4.8 - Transient process of the stand with the delay link turned on and the use of the control program presented in section 4.2.

It should be noted that the quality of regulation is quite good. The dynamic characteristics of the control coincide with the characteristics of a PID controller with a Smith predictor.

The results of testing the fuzzy control algorithm (perceptron) presented in section 3.4 in manual mode (shown in Figure 4.9). The results were obtained under the conditions of a real object (6/8-story, 2-entrance residential building) with the standard automatic control system disabled.

From the graph you can see a fairly quick (30-50 minutes) exit to heating mode. No significant temperature rise during overshoot.

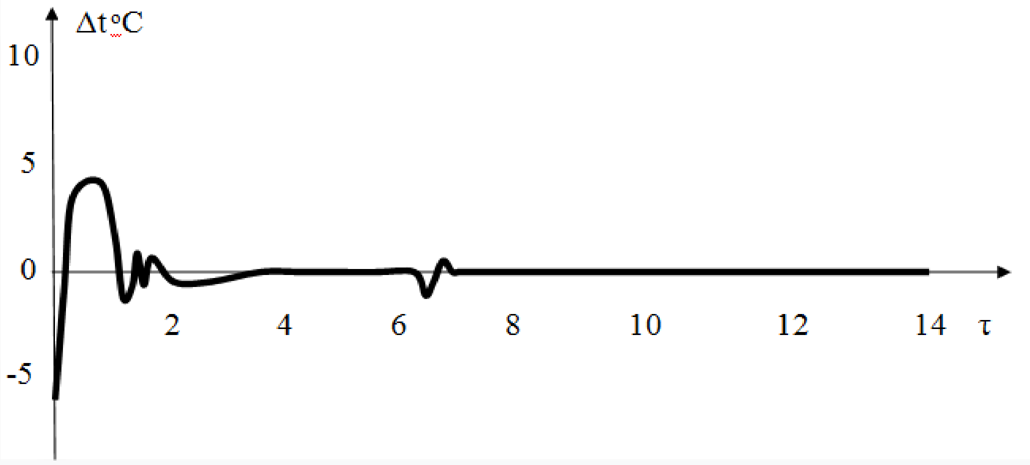


Figure 4.9 - Graph of changes in return network water temperature when using the FL control algorithm.

The transport delay time is on average 12 minutes (10 minutes for the 6-story part and 13 minutes for the 8-story part). The different floors of the building did not allow us to more accurately set the regulator parameters, but, as it turned out, this was not necessary, since the water temperature of the heating system with such different floors had the property of self-levelling, due to the mixing of flows.

Temperature fluctuations during regulation very quickly (within 6-8 regulation cycles) decreased to the value of the sensitivity of the sensor, which made it possible to completely eliminate the operation of the mechanical part of the regulator.

A small disadvantage of this method of regulation can be considered a rather significant stage of “training” of the regulator (5-6 hours), but it should be noted that the regulator

CONCLUSION

The main results of this work can be formulated as follows:

- The main control algorithms used in industry, presented in the work (pure PID controller and PID controller with Smith predictor), cannot be used at all to control objects with large inertia, for example, heating water for a heating system, due to the large transport delay time.

- The developed control method using a perceptron (at the initial stage of regulation) and fuzzification and defuzzification schemes during subsequent regulation make it possible to achieve good results in regulating such a complex object as a heating system (for which the delay time can reach 15-20 minutes) compared to the one currently used time by relay control system.

- Any control algorithms are modeled quite well in simulation systems, such as Simulink, Vissim or Simintech. Programming them in systems for microcontrollers such as Arduino allows us to evaluate their effectiveness in bench conditions.

- The issue of “training” the perceptron (its knowledge base) remained unexplored due to the complexity of the control program for the microcontroller and lack of time. But further work in this direction will solve this problem.

- The developed algorithm for controlling the heating system in the form in which it is presented in this work can be used in practice.

REFERENCES

1. K.J. Åström, and T. Hägglund, PID Controllers: Theory, Design and Tuning // 2nd edition, ISA, 1995.;

2. Sh. Bhagwan, A. Kumar, J.S. Soni. A Review on: PID Controller // International Journal on Recent Technologies in Mechanical and Electrical Engineering;

3. S. Bennett. The past of PID controllers // Annual Reviews in Control. – 2001. – №25. – P. 43–53;

4. KiamHeongAng, Gregory Chong, Yun Li. PID Control System Analysis, Design, and Technology // IEEE Transactions on control systems technology. – 2005. – №13(4). – P.559 – 576;

5. I. Kaya, N. Tan and D. P. Atherton. A refinement procedure for PID controller, Electrical Engineering // Electrical Engineering (Archiv fur Eletrotechnik). – 2006. – №88(3). – P.215-221;

6. M. Kushawah, A. Patra. Tuning PID Controller for Speed Control of DC Motor Using Soft Computing Techniques-A Review. – 2014. – №4(2) – P. 141 – 148;

7. Kharitonov V.L. Time-Delay Systems.LyapunovFunctionals and Martices.Birkhäuser. – 2013. – 312 p.;

8. A. Soukkou, M.C. Belhour, S. Leulmi. Review, Design, Optimization and Stability Analysis of Fractional-Order PID Controller // Int. J. Intel. Syst. Techn. Appl. – 2016. – №8(7). – P. 73-96;

9. Astrom K.J., Hagglund T. Advanced PID control. - ISA - The Instrumentation, Systems, and Automation Society. – 2006. – 460 p.;

10. K. K. Tan, T. H. Lee, R. Ferdous. New Approach for Design and Automatic Tuning of the Smith Predictor Controller // Ind. Eng. Chem. Res. – 1999. – №38. – P.3438-3445;

11. T. H. Lee, Q. G. Wang. Robust Smith-Predictor Controller for Uncertain Delay

12. V. Kumar, B.C. NakraNakra, A. Mittal. A Review on Classical and Fuzzy PID Controllers // International Journal of Intelligent Control and Systems;

13. The Heating Book. 8 stepscontrol of heating systems. – Nordborg: Danfoss a/s, 2021.– 185 p.

14. A.A.Shilin, V.G.Bukreev Study of a three-position temperature controller in a sliding mode of operation. Reports TUCUPa. №1(5), part 2, June 2012, p.251-257.

15. Huang B.Q., Tarik Rashid and M-T. Kechadi. Multi-Context Recurrent Neural Network for Time Series Applications // INTERNATIONAL JOURNAL

OF COMPUTATIONAL INTELLIGENCE. VOLUME 3, NUMBER 1, 2006. p.p. 1304-2386.

16. Mendel J., Mouzouris G. Non-Singleton fuzzy logic systems: Theory and application", IEëEE Trans. Fuzzy Syst., vol. 5. pp. 56-71, Feb. 1997.