DOI: 10.15828/2075-8545-2021-13-4-252-258
Article type: Original article

Development of an intelligent control system for the process of preparation and water transfer in the cooling circuit of an ammonia station

E.A. Muravyova* , **A.V. Kochenkov**
Branch of Ufa State Petroleum Technological University, Sterlitamak, Russia
*** Corresponding author:** e-mail: muraveva_ea@mail.ru

ABSTRACT: Introduction. In the modern socio-economic and geopolitical development of Russia, the development of industry comes to the fore. Among the many industries, ammonia stations play the most important role. **The main regularities of the process of pumping and preparing water.** The process consists of six stages, this article discusses the automation of stages 1 and 2: for water treatment and pumping it out with pumps H1 and H2 from the tank P2. Products in the form of purified water are the most important criteria for subsequent production at an ammonia plant, therefore, increased requirements are imposed on the quality of finished products, including the quality of purification of the water used with the help of nanofilters. The required quality cannot be achieved without control the process in an automated mode. **Development of a neural network.** To control the converters frequency values during the preparation and pumping of water, an artificial neural network must be used. Its development was carried out in the Matlab environment in the Neural Network Toolbox package, input and output data were defined for this, data processing and preparation were performed, as well as the choice of the type and architecture of the neural network. The architecture of the Layer Recurrent neural network, the process of its construction and training in Matlab is described. **Testing of neural networks.** During testing of the Layer Recurrent network for the degree of their training, the smallest error was obtained for 30 neurons in the hidden layer. The proximity to the set values indicates the applicability of the network for controlling the parameters of frequency converters. **Development of the neural network controller model in the Simulink package.** The simulation of the control system in the Simulink package using a neural network controller with the Layer Recurrent architecture is performed. Checking the frequencies of the frequency converters H1 and H2 in Simulink for the level parameters in the tanks and in the tank $LT_{1\_вх}$, $LT_{2\_вх}$, $LT_{3\_вх}$ showed that the object model works correctly, thus, the simulation of the neural network showed that the training was successful. **Conclusion.** As a result of the conducted research, an artificial neural network was developed to control the process of preparing and pumping water in the Matlab environment and a simulation of a neural network in the Simulink package.

KEY WORDS: process, ammonia station, nanofilters, development, neural network.

FOR CITATION: Muravyova E.A., Kochenkov A.V. Development of an intelligent control system for the process of preparation and water transfer in the cooling circuit of an ammonia station. *Nanotechnologies in Construction.* 2021; 13(4): 252–258. Available from: doi: 10.15828/2075-8545-2021-13-4-252-258.

INTRODUCTION

For the process of water preparation and pumping to feed it to the cooling circuit of the ammonia station, it is necessary that the frequency converters of the H1 and H2 pumps work without interruption. The operation of frequency converters is determined by the liquid level in the reservoir P2, in sections 1 and 2, and the level of the liquid in the backwater basin B1.

If the levels in sections 1 and 2 are high, and the level in tank B1 is low, then it is necessary that the values of the frequencies of the converters are high. If the levels in sections 1 and 2 are low, and the level in tank B1 is high, then it is necessary that the values of the frequency converters are low.

This article investigates a method for controlling the processes of water preparation and pumping occurring at stages 1 and 2, precise control of which is necessary for a smooth supply of water for purification to nanofilters, it is the use of a neural network to create a supply controllers that protects nanofilters from failure, thereby providing of their safety system.

The aim of the study is to develop an artificial neural network to control the process of preparing and water transfer in the Matlab R2017b environment and to simulate the neural network in Simulink.

To maintain the stability of the technological process for the preparation and water transfer, it is necessary to control directly the parameters of the frequency converters of the pumps. Based on the research carried out, a neural network was developed, which is used to regulate the parameters of pumps by using frequency converters F1 and F2 [8].

When using a control system based on PID controllers, there is a problem of the lack of compensation for disturbances and taking into account the mutual influence of technological parameters, which leads to instability in the control of the object. [5].

1. Main regularities of the process water pumping and treatment

The technology of water pumping and treatment is based on the collection of stormwater runoff into the tank P1, followed by pumping by pumps to filter columns. The process of water pumping and treatment includes the main stages [7]:

1) water treatment (collection of storm drains, sludge and coarse filtration) from tank P1 to tank P2;

2) pumping water with pumps H1 and H2 from the tank P2;

3) water purification on nanofilters F01 and F02 from small mechanical impurities;

4) water purification by ion exchange method in the filter column F03;

5) water purification on the filter column of de-ironing F04;

6) purification from ammonia by filtration on an ammonia filter F05, which helps to improve the pH concentration of purified water.

As the final products of the production process of water pumping and treatment, purified water is obtained to feed the cooling circuit of the ammonia station.

During flushing, the contaminated water is drained into the sewer.

The process of water pumping and treatment is designed to purify water and feed it into the cooling circuit of an ammonia station. This article discusses the automation of the 1st and 2nd stages of the process.

The process of preparation and water transfer includes [4]:

– tanks (P1, P2) – designed for collection, fine filtration and subsequent pumping to filter columns;
– pumps (H1, H2) – transfer water from the P2 reservoir to the filter columns;
– skimmers (H3, H4) – collect oil slick from reservoir P2;
– filter columns (F01–F05) (5 pieces) – carry out complete water purification.

2. Development of a neural network

To control the values of the converter frequencies during the preparation and water transfer, it is necessary to introduce an artificial neural network.

The development of ANN (artificial neural network) is carried out in the Matlab R2017b environment in the Matlab Neural Network Toolbox [7].

The process of preparing and water transfer for building a neural network model can be conditionally divided into 5 main stages.

At the first stage, the analysis and selection of input data was carried out, which significantly affect the controlled process.

At the second stage, the definition and collection of input data is carried out and the methods of obtaining the output values are selected.

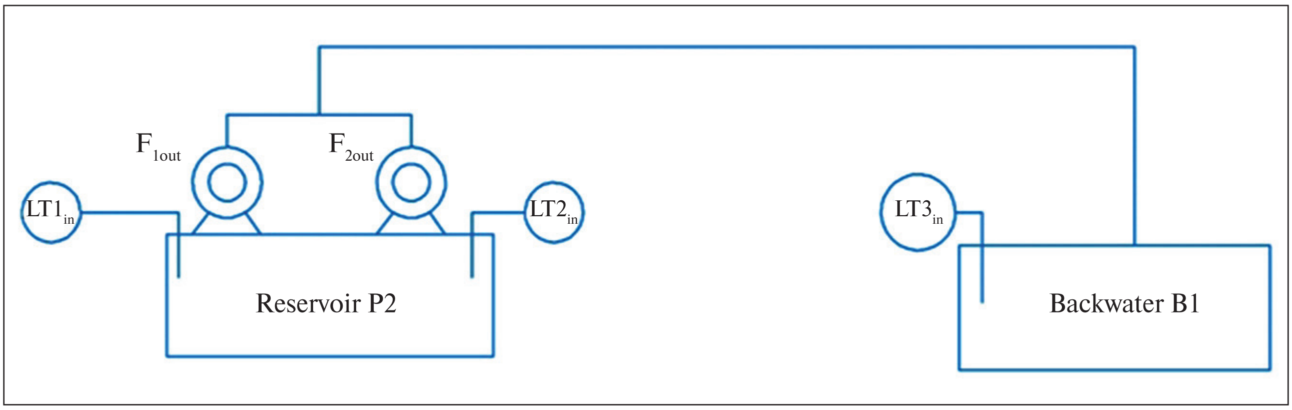

Fig. 1. **Functional diagram of the automation of the process of preparation and water transfer**

The third stage is the choice of the ANN architecture.

At the fourth stage, the network is trained.

At the fifth stage, the effectiveness of the created neural network is checked. [2-5].

2.1. Definition of inputs and outputs

To solve the problem using a neural network, it is necessary to collect data for training. Output parameters - the values of frequencies of frequency converters F_{1out}, F_{2out} vary depending on the input parameters $LT1_{in}$, $LT2_{in}$, $LT3_{in}$ according to the dependencies below. The output was calculated using the formulas:

$$F_{1out} - (LT1_{in}/(LT3_{in}/\ LT1_{in}))\bullet 2; \quad (1)$$

$$F_{2out} - (LT2_{in}/(LT3_{in}/\ LT2_{in}))\bullet 2, \quad (2)$$

where $LT1_{in}$ – water level in section 1 of reservoir P2;
$LT2_{in}$ – water level in section 2 of reservoir P2;
$LT3_{in}$ – water level in the circulating water tank B1;
F_{1out} – first pump power;
F_{2out} – second pump power.

If the levels in sections 1 and 2 are high and the level in tank B1 is low, then it is necessary that the values of the frequency converters are high. In the event that the levels in sections 1 and 2 are low, and the level in tank B1 is high, then it is necessary that the values of the frequencies of the frequency converters are low [10].

Next, let's prepare the data for training the neural network.

3 values will be fed to the input of the neural network:
1. Water level in section 1 $LT1_{in}$ of tank P2 (0÷100) %;
2. Water level in section 2 LT_{2_in} of tank P2(0÷100) %;
3. Water level in backwater basin LT_{3_in} B1(0÷100) %;

At the output, the neural network must calculate:
1. The number of revolutions of the pump motor F1(0÷2000) r/s;
2. The number of revolutions of the pump motor F2(0÷2000) об/ r/s.

2.2. Data processing and preparation

Using formulas (1–2), let's create a table with the training set. It consists of 1044 examples.

2.3. Choosing the type and architecture of a neural network

During the analysis, it was decided to select Layer Recurrent neural networks (recurrent neural networks). The selection criteria were as follows: high accuracy of the output values due to feedbacks of the selected architecture, nonlinear structure of the selected neural network architecture, a large number of sources for it.

Table 1
The training set

№	Inputs			Outputs	
	$LT1_{in}$, %	$LT2_{in}$, %	$LT3_{in}$, %	F_{1out}, r/s	F_{2out}, r/s
1	1	1	5	0	0
2	1	1	10	0	0
3	1	1	15	0	0
4	1	1	20	0	0
5	1	1	25	0	0
6	1	1	30	0	0
...	...	...	...	...	...
1039	100	100	75	266	266
1040	100	100	80	250	250
1041	100	100	85	235	235
1042	100	100	90	222	222
1043	100	100	95	210	210
1044	100	100	100	200	200

There is no unambiguous methodology for choosing the number of hidden layers and neurons in them, and the question of how successful this or that choice is, is often decided on the basis of the experimental results of learning and testing ANN [9].

It is necessary to select the number of neurons in the hidden layer. Choosing the right amount is a very important step [2]. Too few and the network will not be able to learn. Too much will lead to an increase in the training time of the network to a virtually unrealistic value. There is no easy way to determine the required number of elements in a hidden network layer. In this case, the required number of neurons in the hidden layer for the smallest error of the Layer Recurrent neural network was experimentally established – 30 [14].

The network will be trained using a modified error backpropagation algorithm: trainFcn = 'trainbr'.

We set the maximum number of training epochs, which determines the number of epochs (time interval) after which training will be terminated: epochs = 1000.

Let's choose the number of epochs between impressions equal to twenty five: show = 25

We set the criterion for the end of training – the deviation value at which the training will be considered complete: goal = 0.

2.4 Layer Recurrent neural network architecture description

Layer Recurrent neural networks are networks where connections between elements form a directional sequence. This makes it possible to process data in time or sequential spatial data chains. The training of recurrent neural networks is carried out by the method of back propagation of the error, which makes it possible to train the weight coefficients with repeated training of the network, which in turn increases the efficiency of training the ANN [13].

2.5 Building and training a neural network in Matlab

We will implement and train a neural network of the Layer Recurrent architecture in Matlab. For this we use the nntool instrument. Let's call the input data of the neural network input, and the output data – output. Let's select the Layer Recurrent neural network. The uploaded data will be displayed in the working window "Workspace".

The entrance to the "Neural network training" tab is carried out using the nntool command [7]. A signal x is received at the input of the neural network, which consists of the following parameters: the water level in the reservoir P2 of section 1 LT_{1in}, the level in the reservoir P2 of section 2 LT_{2in}, the level in the backwater basin B1 LT_{3in}. An adder "+" multiplies each input b_i by the weight w_i and adds the weighted inputs. Then the value passes through the activation function of the corresponding layer and the outputs are calculated: the powers of the frequency converters F_{1out}, F_{2out}.

Using the NNTool command, the entrance to the "Neural network training" tab will be implemented. In the Input Data work window, the input values "Input" are entered. In the work window Target Data (target data) enter the output values "Output".

2.6 Layer Recurrent architecture neural network

Let's analyze the following Layer Recurrent neural network. 30 neurons were selected in the hidden layer.

At the end of the NN learning process, a window for the completion of the learning process appears, which displays the number of iterations performed – 1000, the time spent on training – 00:01:07 h, the value of the root mean square error – 0.18887, the value of the gradient – 0.27445, the value of regularization – 0.01 and the

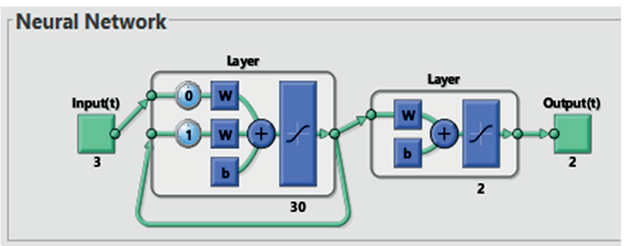

Fig. 2. **Training a neural network**

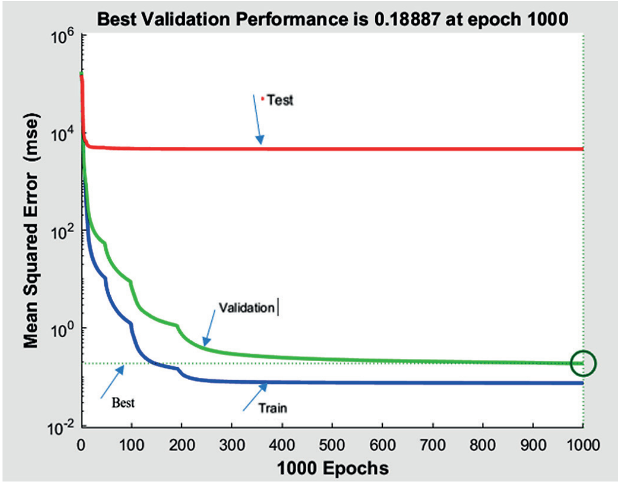

Fig. 3. **Mean square error**

frequency of deviations of the obtained values from the specified error is 0.

The change in the real value from the true we can be observe on the training graph of the Layer Recurrent neural network in the training completion window (Figure 3). From the training graph of the Layer Recurrent neural network, we can see that the mean square error value of 0.18887 was achieved in 1000 epochs, which is less than the specified one – 0.315.

Further, you can see how the gradient and the learning coefficient changed during the training of the neural network (Figure 4). The gradient of the functional of the learning error by the weights of the Layer Recurrent neural network is shown in the gradient graph (Figure 4). The weights of the neural network are considered as arguments of the Layer Recurrent neural network, where a gradual movement takes place to lower and lower points in the plane of the graph in order to find the minimum. On the

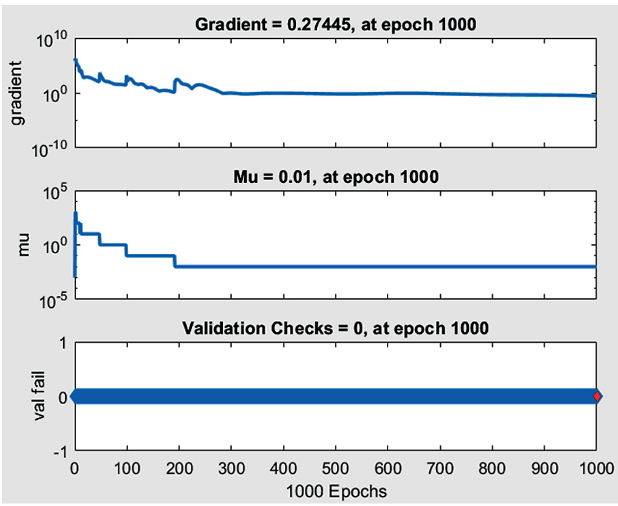

Fig. 4. **Graphs of network training**

2021; 13 (4):
252–258

gradient graph when training the Layer Recurrent neural network, you can observe abrupt transitions to find the best value for adjusting the values of the training sample. The gradient value (Figure 4) for the Layer Recurrent neural network equal to 0.27445 shows the sum of all adjustments for the variables that were maximally higher than the error rate.

The change in the regularization variable when training the Layer Recurrent neural network can be observed on the Mu graph (Figure 4). The regularization parameter is changed using the method we have chosen – Levenberg-Markart. To adjust the values of the training sample of the Layer Recurrent neural network and retrain the neural network, a range of numerical values is used – this is called regularization. In Figure 4, you can see that this number takes on a value of 10^{-3}, and we can conclude that the rate of the value is small, so it can be neglected.

The frequency of deviations of the obtained values from the prescribed error is shown by Validation checks (Figure 4). Figure 4 shows that:

– in the range from 0 to 1000 learning epochs, the frequency of deviations does not change and equals 0.

After analyzing the Validation checks graph, we can conclude that in the process of training the Layer Recurrent neural network for the smallest error, it is better to stop training the Layer Recurrent neural network at 1000 epochs. Since deviations no longer occur after 1000 epochs, it is advisable to stop training the neural network.

The final value of 0, which we observe on the Validation checks graph, shows that the frequency of deviations of the Layer Recurrent neural network reaches the minimum value when the 1000 epoch is reached.

3. Neural network testing

To test neural networks for the degree of their training, we supply 3 values to the input in the main Matlab window [15–16].

3.1 Testing the neural network of the Layer Recurrent architecture

We introduce a new variable xx = [100; 100; 100] $LT1_{in}$ = 100, $LT2_{in}$ = 100, $LT3_{in}$ = 100. We introduced a new variable xx to check the output values (frequency converters) [6].

Next, let's simulate and print the output values.

1. For 30 neurons in the hidden layer

To test the neural network, take the test data (Table 2).

Test results: received output values of frequencies of frequency converters F_{1out}, F_{2out}.

F_{1out} – 199.75;

F_{2out} – 199.74.

During testing of the Layer Recurrent network, the smallest error was obtained for 30 neurons in the hidden layer [4]. Initial data (200; 200). The proximity to the specified values (199.75; 199.74) indicates the applicability of the network. In the future, it can be used to control the parameters of frequency converters.

Table 2

Verification data for output values (for 30 neurons)

Input			Output	
$LT1_{in}$, %	$LT2_{in}$, %	$LT3_{in}$, %	F_1, r/s	F_2, r/s
100	100	100	200	200

4. Development of a neural network controller model in the Simulink package

Figure 5 shows a general view of the Layer Recurrent neural network, where the Custom Neural Network block denotes the artificial neural network itself. Into the Constant block we enter the input values LT_{1_in}, LT_{2_in}, LT_{3_in}, and the block y1 displays the output values of F_{1out} and F_{2out}.

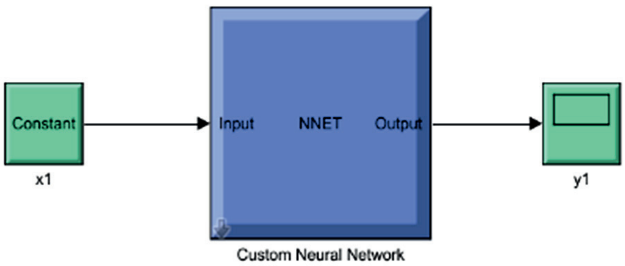

Fig. 5. **General view of a neural network in Simulink**

Figure 6 shows the general architecture of Layer Recurrent. From the Input block, the signals LT_{1_in}, LT_{2_in}, LT_{3_in} are sent for analysis to the Process Input 1 block, then the input signal goes to the hidden layer Layer 1. Layer 2 accepts the Layer 1 output and calculates the outputs F_{1out} and F_{2out}.

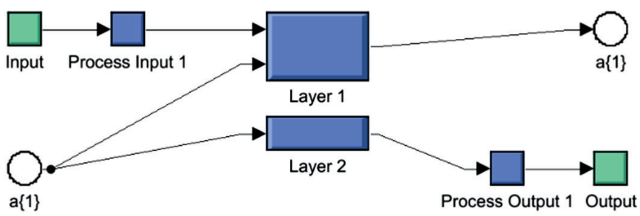

Fig. 6. **General architecture of Layer Recurrent**

Figure 7 shows the hidden layer of the neural network Layer 1, where the adder "+" multiplies each input b_i by the weight w_i and sums the weighted inputs.

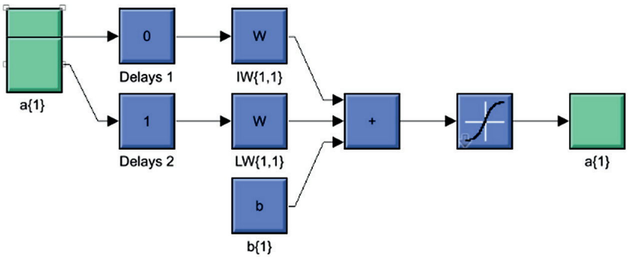

Fig. 7. **Hidden layer of the neural network Layer 1**

Figure 8 shows the output layer of the neural network Layer 2, the value from Layer 1 passes through the activation function of the corresponding output layer of Layer 2 and the outputs F_{1out} and F_{2out} are calculated.

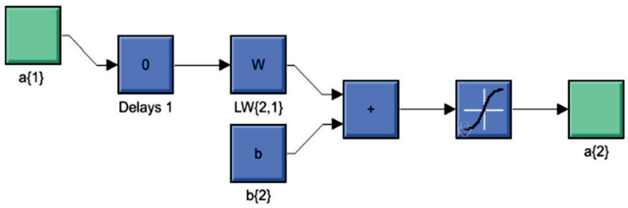

Fig. 8. **The output layer of the neural network**

Let's check the frequency converters F_{1out} and F_{2out} in Simulink for LT_{1_in}, LT_{2_in}, LT_{3_in} [100; 100; 100] then change the input values to [4; 4; 100] (Figure 9). Figure 9 displays the F_{1out} and F_{2out} values for the input values LT_{1_in}, LT_{2_in}, LT_{3_in}.

CONCLUSION

The article substantiates the feasibility of developing an intelligent control system for the parameters of frequency converters of pumps, implemented using a neural network Layer Recurrent. A model for controlling the parameters of frequency converters is presented, taking into account the multiple relationships between the parameters of the technological process and control signals.

As a result of training, a neural network was obtained that generates output signals with a minimum error. The initial data of the frequency converters were chosen equal to 200 r/s. The proximity of the test result to the specified values (199.75; 199.74) indicates the applicability of the network for controlling the parameters of frequency converters.

The simulation of the control system in the Simulink package using a neural network controller using the Layer Recurrent architecture has been performed. Checking the frequencies of frequency converters F_{1out} and F_{2out} in Simulink for the level parameters in reservoirs and in the tank LT_{1_in}, LT_{2_in}, LT_{3_in} [100; 100; 100] showed that the object model is working correctly, that is, the developed neural network for the neurocontroller is trained correctly. Then we changed the input values to [4; 4; 100], which are new for the neural network. The model with them also showed correct process control.

Thus, an artificial neural network has been developed, trained and tested, which effectively controls the process of preparation and water tranfer.

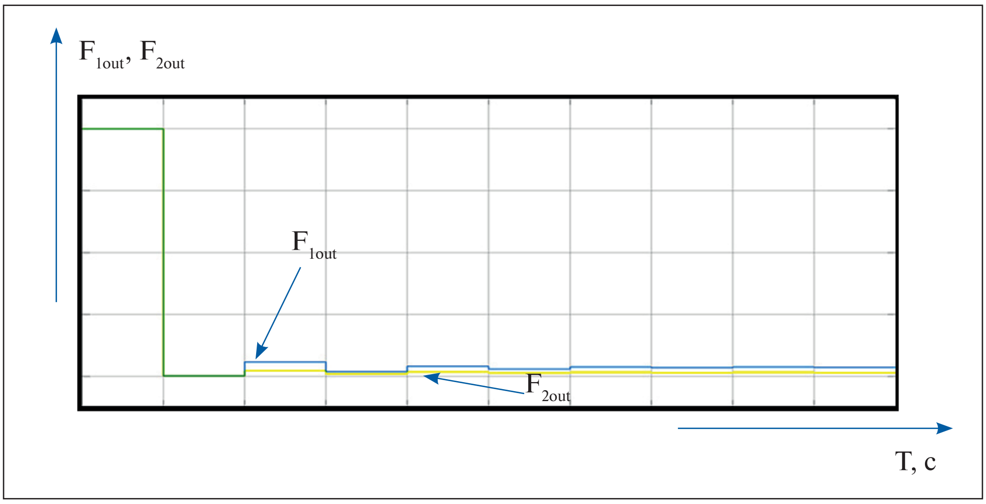

Fig. 9. **The output layer of the neural network F_{1out} and F_{2out}**

REFERENCES

1. Muravyova E., Popkov V. Development and Research of a Dynamic Flow Laboratory Bench Model. In: *Advances in Intelligent Systems Research 7th Scientific Conference on Information Technologies for Intelligent Decision-Making Support (ITIDS)*. 2019. P. 1–6.

2. Muravyova E., Gabitov R. Economic Features to Optimize the Catalyst Calcinations Process. In: *International scientific multi-conference on industrial engineering and modern technologies (FAREASTCON)*. 2018. P. 1–8.

3. Muravyova E., Sharipov M., Gabitov R. SCADA-System Based on Multidimensional Precise Logic Controller for the Control of a Cement Kiln. In: *International scientific multi-conference on industrial engineering and modern technologies (FAREASTCON)*. 2018. P. 1–9.

4. Muravyova E., Bondarev A. Method for Increasing the Speed and Reducing the Error of Multidimensional Precise Logic Controller. In: *International scientific multi-conference on industrial engineering and modern technologies (FAREASTCON)*. 2018. P. 1–11.

5. Muravyova E., Sharipov M. Method of optimal parameters control in three-phase separator using fuzzy controller. In: *Proceedings of the International conference actual issues of mechanical engineering (AIME)*. 2018. P. 1–8.

6. Muravyova E., Fedorov S., Bondarev A. Control Systems with Pulse Width Modulation in Matrix Converters. In: *International conference on mechanical engineering, automation and control systems (MEACS)*. 2017. P. 1–7.

7. Muravyova E., Bondarev A., Sharipov M., Galiaskarova G., Kubryak A. Power consumption analysis of pump station control systems based on fuzzy controllers with discrete terms in iThink software. In: *International conference on mechanical engineering, automation and control systems (MEACS)*. 2017. P. 1–8.

8. Muravyova E., Azanov A., Enikeeva E. Modeling power consumption by pump station control systems based on fuzzy controllers with discrete terms. In: *Proceedings of the international conference: aviamechanical engineering and transport (AVENT)*. 2018; p. 1–6.

9. Muravyova E., Sharipov M., Radakina D. Method of fuzzy controller adaptation. In: *Proceedings of the international conference actual issues of mechanical engineering (AIME)*. 2017. P. 1–7.

10. Muravyova E., Bondarev A. Fuzzification Concept Using the Any-time Algorithm on the basis of Precise Term Sets. In: *International conference on industrial engineering, applications and manufacturing (ICIEAM)*. 2017. P. 1–9.

11. Muravyova E., Sharipov M. Two Fuzzy Controller Synthesis Methods with the Double Base of Rules: Reference Points and Training Using. In: *International conference on industrial engineering, applications and manufacturing (ICIEAM)*. 2017. P. 1–7.

12. Muravyova E., Kayashev A., Sharipov M., Emekeev A., Sagdatullin A. Verbally Defined Processes Controlled by Fuzzy Controllers with Input/Output Parameters Represented by Set of Precise Terms. In: *International conference on mechanical engineering, automation and control systems (MEACS)*. 2014. P. 1–8.

13. Yoshiko H., Shunji U., Taiko K. Evaluation of artificial neural network classifiers in small sample size situations. In: *Pros. Int. It. Conf. Neural Networks*. 2019. P. 1–6.

14. Wasserman F. *Brain-computer equipment*. Transl. from Eng. Moscow: Mir; 1992.

15. Hrycej T. *Neurocontrol: Towards An Industrial Control Methodology*. John Willey & Sons. 2018. 380 p.

16. Istvan Szite, Andras Lorincz. *Simple algorithm for recurrent neural networks that can learn sequence completion (IJCNN)*. 2019. P. 1–6.

INFORMATION ABOUT THE AUTHORS

Elena A. Muravyova, Dr. Sci. (Eng.), Professor, Head of Department «Automated Technological and Information Systems», Branch of Ufa State Petroleum Technological University, Sterlitamak, Russia,
ORCID: https://orcid.org/0000-0002-7118-5570, e-mail: muraveva_ea@mail.ru

Alexander V. Kochenkov, Student of the Department "Automated technological and information systems", Branch of Ufa State Petroleum Technological University, Sterlitamak, Russia,
ORCID: https://orcid.org/0000-0002-5346-6491, e-mail: ko4enkov.alex@yandex.ru

Authors declare the absence of any competing interests.

Received: 30.06.2021.
Revised: 19.07.2021.
Accepted: 22.07.2021.