*Adv Syst Sci Appl* 2023; **03**; 127-138
Published online at https://ijassa.ipu.ru.

# Quality Control of a Real-Time Flight Experiment Using Neural Networks

Vasiliy Akhrameev[1], Eugeniy Tsvetkov[1], Alexander Pashchenko[2*]

[1)] *Moscow Institute of Physics and Technology, Moscow, Russia*

[2)] *V.A. Trapeznikov Institute of Control Sciences of RAS, Moscow, Russia*

**Abstract**: The paper presents some investigations on automatic processing of flight information. The aim of this investigation is to reduce the cost and timing required for aviation equipment testing. The algorithms are developed for recognizing various test modes as well as evaluating the correctness of their implementation. While developing the algorithms a multilayer perceptron and a Kohonen network were used as basics. The results of experiments are presented when applying these algorithms for real testing process.

***Keywords***: automatic processing of flight information, test modes, flight experiment, multilayer perceptron, Kohonen network

## 1. INTRODUCTION

One of the ways to reduce the time and cost of flight tests is the use of automatic flight information processing tools. This is especially important, since the amount of information collected during the flight experiment is growing more and more and requires a lot of time and human resources for secondary data processing.

Modern methods of quantifying the stability and controllability of an aircraft are based on a comprehensive analysis of the records of its movement obtained in flight when the pilot performs special control maneuvers [1], [2], which are one of the main elements of the flight task of a test pilot. For qualitative and quantitative assessment of the dynamic stability and controllability of the aircraft, the following test control modes are usually performed:

- impulse deviations of control surfaces;
- stepwise deviations of control surfaces;
- two-way deviations of control surfaces.

To carry out secondary processing of experimental data in real time, it is necessary to recognize the test modes and, based on the recognition results, include certain secondary processing algorithms. This task should be solved completely automatically and as secretly as possible from the operator who solves the main tasks of controlling the flight experiment.

The main feature of the flight modes during the flight experiment is their determinacy, since the flight task is negotiated in advance before departure, the required values of the deviation of the aircraft controls are specified. Typically, a test pilot is specifically trained to perform the modes. Then the problem of recognition is reduced to the well-known problem of coherent reception in statistical radio engineering

## 2. PROBLEM STATEMENT

In order to conduct secondary processing of the results of the flight experiment, it is required to develop an algorithm for recognizing test modes and assessing the correctness of its implementation.

[*] Corresponding author: paschenko_alex@mail.ru

The work sets the following tasks:

1) development of an algorithm for finding flight modes corresponding to a given reference mode, based on the use of neural networks of two types: a multilayer perceptron and a Kohonen network;
2) development of an algorithm for training networks;
3) application of the resulting algorithm for real-time recognition;
4) investigation of the recognition system for the stability of decision-making when submitting noisy and highly distorted records.

## 3. MAIN TASKS OF RECOGNITION SYSTEM CONSTRUCTION

Each recognition system is adapted to recognize only this type of objects or phenomena. Let's consider the main tasks that arise in the process of designing and building recognition systems.

The first task is to study in detail the objects that will be recognized by the system being created. The main purpose of this task is to find out the features of the studied objects and find out which features unite and which distinguish objects from each other.

The second task is to divide the entire set of recognizable objects or phenomena into separate classes. To do this, you need to choose the correct classification method based on the requirements that apply to the recognition system itself. The main criterion here is a set of decisions made when recognizing unknown objects and phenomena.

In the third task, the goal is to compile a dictionary of features used both for a priori description of the class and for classification of each unknown object submitted to the recognition system input.

The dictionary can only include features about which information can be obtained sufficient to describe classes in the language of these features. It is impractical to include all the signs in the dictionary, even if the information about them is received in full.

When designing a dictionary in the language of features, classes are described and after selecting recognition algorithms, the informativeness of each feature is evaluated [3]. Thus, it is possible to weed out the least useful signs. After that, taking into account the restrictions imposed on the creation of technical means of obtaining information, a final decision is made on the composition of the dictionary of features for the operation of the recognition system.

The fourth task involves the description of classes of objects based on the resulting dictionary of features. The task of describing classes is as follows. Let the dictionary contain an ordered set of features $X_1, \ldots, X_N$. This set of parameters is a vector $X = \{X_1, \ldots, X_N\}$ defined in the feature space of the recognition system. Each point of this space represents recognizable objects.

The whole set of objects is divided into classes $\Omega_1, \ldots, \Omega_m$, each class in the feature space corresponds to a region $D_i, i = 1, \ldots, m$, so that a point in the feature space representing an object that has features $X_1^i, \ldots, X_N^i$ and belongs to the class $\Omega_i$ belongs to the region $D_i$. It is also required to construct separating functions $F_i(X_1, \ldots, X_N), i = 1, \ldots, m$, having the following property – if the object belongs to the class $\Omega_i$, then the value of $F_i(X_1^i, \ldots, X_N^i)$ should be the largest. The separating function $F_i$ must take the largest value for all objects $X^i$ of the class $\Omega_i$.

$$F_q(X^q) > F_g(X^q), q, g = 1, \ldots, m, q \neq g. \tag{1}$$

Thus, in the feature space of the recognition system, the boundary of the partition, called the decisive boundary between the regions $D_i$ corresponding to the classes $\Omega_i$, is expressed by the equation

$$F_q(X) - F_g(X) = 0 \tag{2}$$

The processing of information about recognized objects and the description of classes is a very large and time–consuming part in the task of creating a recognition system associated with a detailed study of the features of objects.

The fifth task is to develop recognition algorithms that determine the class to which the recognized object belongs. Recognition methods are based on comparing some measure of proximity or similarity of the classified object with each class. Thus, if some measure of the proximity of a given object to any class $\Omega_i$ exceeds the measure of proximity to other classes, then a decision is made on whether this object belongs to the class $\Omega_i$

The sixth task is understood as the development of additional methods for controlling the recognition system to optimize the functioning and maximize the selected criterion for the quality of the system.

The selection of criteria by which the effectiveness of the system will be evaluated is the last seventh task. Performance indicators, such as recognition probability, decision-making time and required costs, are determined based on experimental studies of the recognition system using physical and mathematical models.

These seven tasks fully describe the stages of the recognition system development. In this paper, the recognition system is based on the operation of two types of neural networks: a multilayer perceptron and a Kohonen network [3].

## 4. RELEVANCE OF THE CHOICE OF NEURAL NETWORKS FOR CREATING A RECOGNITION SYSTEM

Artificial neural network (INS) is a mathematical model, as well as its software or hardware implementation, built on the principle of organization and functioning of biological neural networks – networks of nerve cells of a living organism. This concept arose when studying the processes occurring in the brain, and when trying to simulate these processes. The first such attempt was the neural networks of W. McCulloch and W. Pitts [4], [5].

When performing a flight experiment, each maneuver corresponds to a certain input and output image. Obtaining the characteristics of a system from a set of input and output signals is a nonlinear task. As described above, algorithms based on the use of multilayer perceptrons are suitable for such tasks.

Kohonen networks are used to solve the recognition problem. Kohonen networks are a type of networks that learn "without a teacher". The network consists of one layer of adjustable weights (Fig. 1) and functions in the spirit of a strategy according to which the winner takes everything, i.e. only one neuron is excited, the remaining outputs of the layer are suppressed, the outputs of the network are:

$$y_j = \sum_i w_{ij} x_i \, , \exists k\colon y_k = 1, y_{j\neq k} = 0. \tag{3}$$

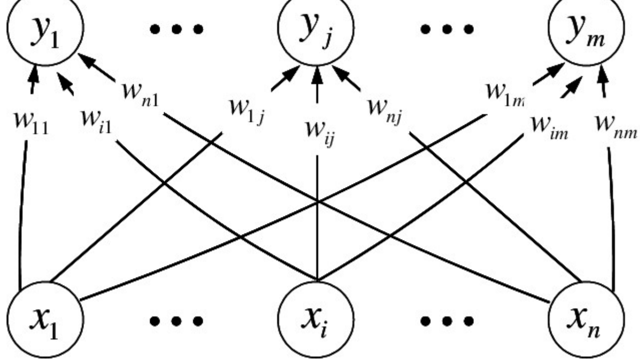



**Fig. 1.** Kohonen Network

The Kohonen network classifies input vectors into groups of similar ones, adjusting the weights so that input images belonging to the same class will activate the same output neuron $y_j$.

It is essential to note that all input vectors $\{x^n\}$ must be normalized.

At each training step, the output neuron with the maximum value is selected. Its weights are adjusted as follows:

$$w^{t+1} = w^t + \mu(x - w^t) \tag{4}$$

where $w^{t+1}$ – is the new weight value.

Adjusting the weights in a similar way reduces to minimizing the difference between the input vector and the vector of weights of the selected neuron.

At the end of training, neurons corresponding to two close classes can be at a significant "distance" from each other. In order for them to be located side by side, you can use a learning algorithm that changes the weights of not only the winning neuron, but also the weights of neighboring neurons.

We introduce a function having a maximum equal to one at zero and decreasing to zero at infinity. A typical example of such a function satisfying the requirements is the Gaussian function:

In this work, a network with output neurons arranged in a row is used. Therefore, the algorithm can be simplified to a discrete model in which, in addition to the winning neuron, the weights of only two neighboring neurons change:

$$w_{j\pm1}^{t+1} = w_{j\pm1}^{t} + 0.1 \times \mu\left(x - w_{j\pm1}^{t}\right) \tag{5}$$

Thus, the output neurons corresponding to one class of images will be adjacent.

Compared to a multilayer perceptron, the Kohonen network is resistant to noise of small amplitude, which is always present in real data obtained in a flight experiment.

As a result, the flight mode search algorithm uses a combination of two networks. The first model, Kohonen's neural network, is used to compress the input vector. The second network, the multilayer perceptron, determines how much this mode corresponds to a certain reference image.

## 5. RESULTS OF MATHEMATICAL MODELING

### *5.1. Setting up a numerical experiment*

In this work, a computer simulation of the giving of elevators was carried out. The longitudinal channel of the aircraft movement in this case is described by the equations

$$I_z\dot{\omega}_z = M_z^{\alpha}\alpha + M_z^{\omega_z}\omega_z + M_z^{\delta_{\text{в}}}\delta_e, \tag{6}$$

$$\dot{\alpha} = \omega_z - \left(n_y - \cos\alpha\right)g/V, \tag{7}$$

$$n_y = \left(C_y^{\alpha}\alpha + C_y^{\delta}\delta\right)qS/mg. \tag{8}$$

In all simulations, the time values of four parameters were recorded: angle of attack – $\alpha$, angular velocity – $\omega_z$, vertical acceleration – $n_y$, elevator deflection – $\delta_e$.

The resulting recordings were also overlaid with noise with a relative amplitude of 5%. Fig. 2 shows an example of the result of computer simulation of the parameters of the longitudinal motion of the aircraft in the test mode when performing an altitude control.

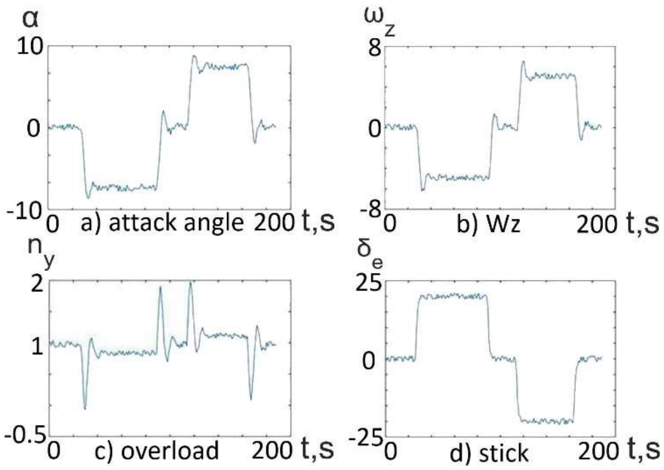



**Fig. 2.** Results of computer simulation of longitudinal motion

### *5.2. Description of recognition algorithms and numerical experiment*

A characteristic record of the flight experiment when performing a control maneuver with the elevator cottages in the time interval is shown in Fig. 3.

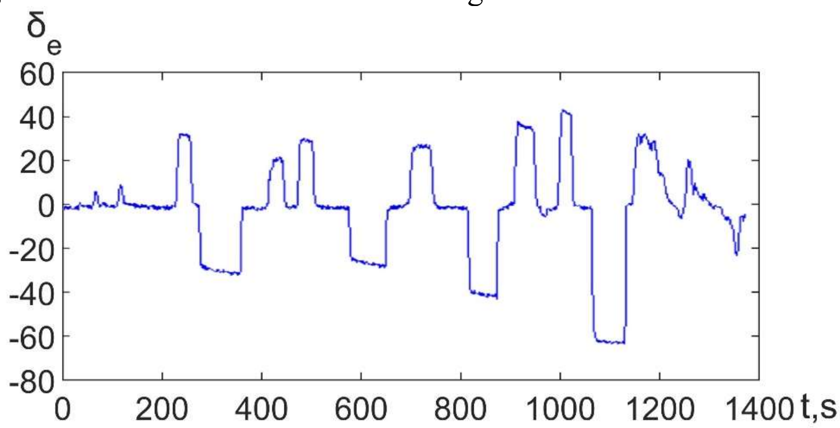



**Fig. 3.** Cottages by the elevator

To train Kohonen neural networks, whose task is to recognize the beginning and end of the test mode, it is necessary to allocate separate intervals at which the elevator is given (Fig. 4).

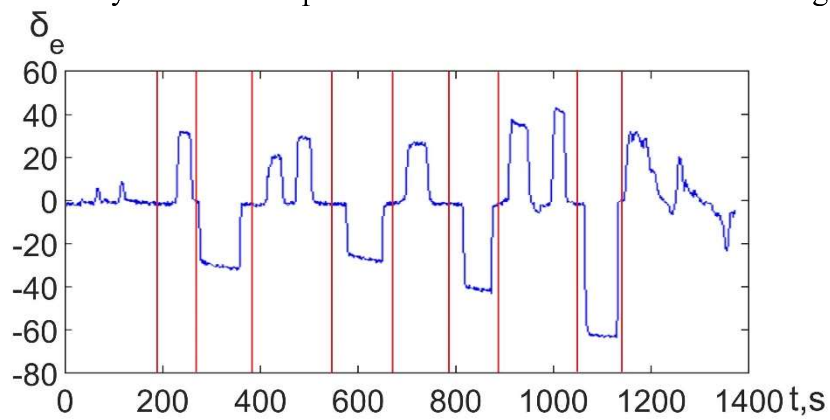
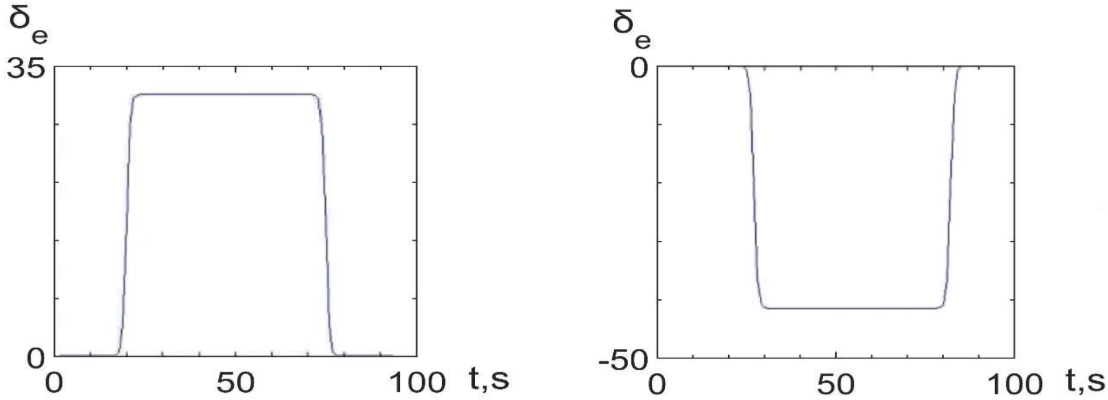



**Fig. 4.** Intervals at which the steering wheel is given

The resulting set of intervals is modeled by reference modes, which can be divided into two groups. In the first group there are modes equivalent to the one shown in Fig. 5a, in the second group – the one shown in Fig. 5b.




a) The first type of standard b) The second type of standard

**Fig. 5.** Reference modes of driving

Also, it is worth adding intervals to the set of training examples in which there are no images of the control maneuver (Fig. 6).

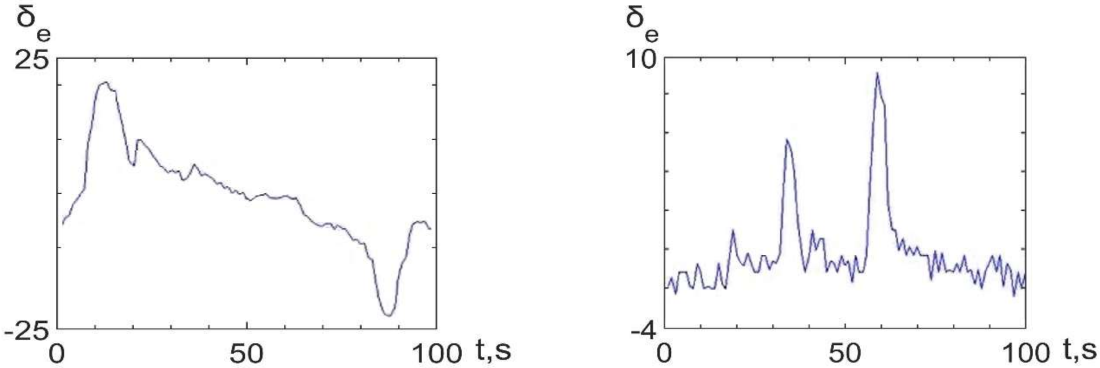



**Fig. 6.** Intervals in which there are no images of the control maneuver

A similar division into sections of the beginning and end of the test mode is performed for all flight parameters that will be used for recognition.

It is important to note that the dimension of the input vector of the Kohonen network is set constant, and the number of points on the interval where the image of the control maneuver is present can vary greatly depending on the duration of the mode and the frequency of sensor polling. Therefore, before submitting a vector to the network input, it is necessary to interpolate this vector into a vector of a given dimension that coincides with the dimension of the network inputs. In this paper, all Kohonen networks used have 50 input and 10 output neurons.

Since the image record may contain areas with negative values, the minimum value is subtracted from all values in the interval. All intervals are normalized to one and the Kohonen network training stage is started.

After training, the output neurons of the network are divided into 3 groups. In the first group there are neurons with the maximum value for the mode shown in Fig. 5a, in the second group there are neurons with the maximum value for the mode shown in Fig. 5b. The remaining neurons form a third group. The values of these neurons are maximal at the intervals shown in Fig. 6.

Table 1 below shows examples of values for the output layer of neurons after the images depicted earlier are fed into the input layer of the trained network.

**Table 1** Examples of values for the output layer of neurons

| | |
|---|---|
| Fig. 5a | 0.1338 0.1155 0.1115 0.2016 0.5873 0.7659 0.8697 0.9336 0.7043 0.7527 |
| Fig. 5b | 0.9017 0.9701 0.9864 0.8886 0.5594 0.4452 0.2485 0.1331 0.3088 0.2726 |
| Fig. 6 | 0.4887 0.4241 0.4120 0.6095 0.8218 0.9092 0.8090 0.7291 0.6543 0.6737 |

As can be seen, in the first case, neurons are mainly excited at numbers 7–10, in the second at numbers 1–4. This result allows us to form a rule for searching and selecting flight recording sites at the primary stage of the algorithm. At this stage, a flight recording section is fed to the input of the Kohonen neural network, in which the experimental mode is presumably located. This section is selected by a "sliding" window moving along the record. The size of the window depends on the type of maneuver performed, and varies within the specified limits. These limits must be set by the operator on the basis of the flight task set before the flight. If any of the neurons takes a value greater than a certain critical value and corresponds to a group of neurons responding to the beginning or end of the mode, then the recognition system stores the corresponding values of the start or end time of the submitted interval.

Further, the recognition system, based on the information received about the start and end time of the found mode, allocates entire intervals at which the maneuver was performed. An example of such a vertical overload recording interval is shown in Fig. 7.

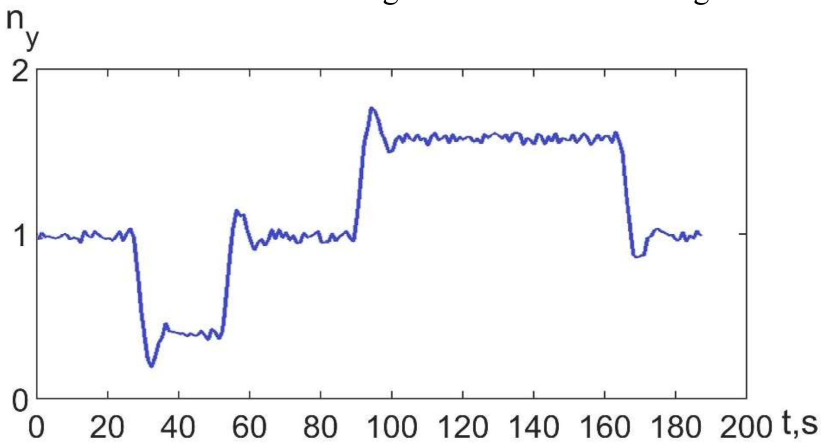



**Fig. 7.** Recording of vertical overload

The arrays obtained earlier already contain the time values corresponding to the beginning and end of the maneuver shown in Fig. 7, so the following algorithm can be applied. Denote the previously obtained arrays as $Reg1(j,i)$ and $Reg2(j,i)$, where the array $Reg1$ contains time intervals corresponding to the class of images shown in Fig. 5, the array $Reg2$ contains the corresponding values related to the class of images shown in Fig. 6. Each $j$-th element of the arrays contains 2 values. We are interested in the value $Reg1(j,[t_{begin},t_{end}])$ and $Reg2(j,[t_{begin},t_{end}])$ , provided

$$\left|Reg1(j,t_{end}) - Reg2\left(k,t_{begin}\right)\right| < T \tag{9}$$

where $T$ is the characteristic time between the execution of the two parts of the desired mode.

The found modes should then be evaluated for the quality of performance by the test pilot. In this paper, a multilayer perceptron is used for this purpose, since the principle of operation and training of this type of neural networks allows us to introduce a fairly free method for evaluating the mode.

As mentioned earlier, Kohonen networks are excellent for compressing data obtained during a flight experiment. The principle of operation of this network, based on the method of selecting the main components, allows you to separate all vectors entering the network by components that distinguish one mode from another.

To train the network, you can use both mode cuts from real flight recordings and simulated ones. In this paper, the second option was used. To create a training set, 2000 modes were modeled, differing in quality when performing elevator handlebars. An example of a correctly executed mode of giving the steering wheel in time is shown in Fig. 8.

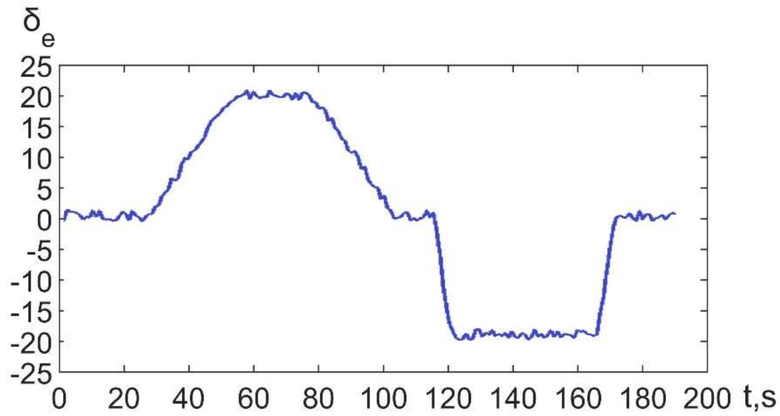



**Fig. 8.** Correctly executed driving mode

If the mode is performed qualitatively, then sharp changes in the angular velocity are observed (Fig. 9).

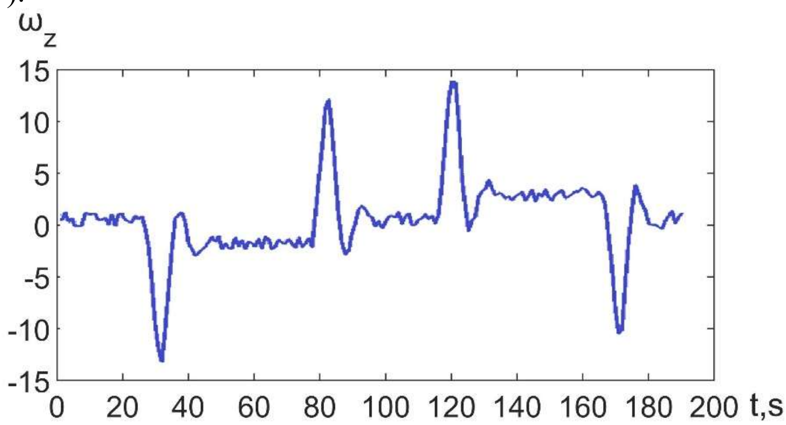



**Fig. 9.** Changes in angular velocity $\omega_z$

With poor-quality execution of the mode (Fig. 10), a smooth change in the angular velocity occurs (Fig. 11).

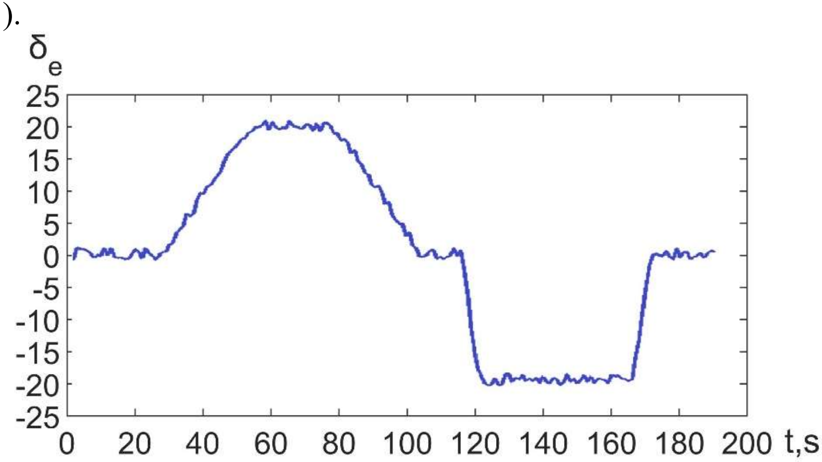



**Fig. 10.** Poorly executed mode

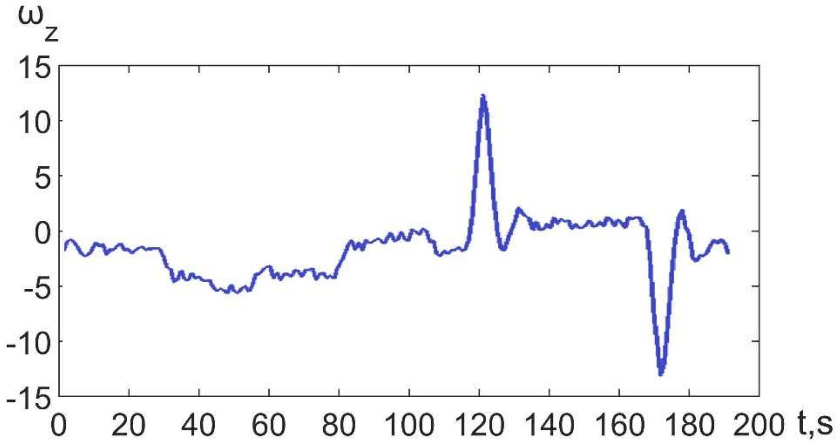



**Fig. 11.** Smooth change of angular velocity

In this numerical experiment, a Kohonen compression network with an input vector dimension of 100 points was trained, and the output vector had a dimension of 30 points. In other words, as a result of training, the network created an internal structure of 30 code vectors, a measure of proximity to which the input vector is encoded.

Each vector supplied to the input of the compression network is smoothed by the moving average method. Let there be an original vector of the mode image $y = \{y_1, \ldots, y_n\}$, then the smoothed vector $y = \{y_1, \ldots, y_n\}$ is obtained according to the following rule

$$yy_1 = y_1, \tag{10}$$

$$yy_2 = \frac{(y_1 + y_2 + y_3)}{3}, \tag{11}$$

$$yy_3 = \frac{(y_1 + y_2 + y_3 + y_4 + y_5)}{5}, \tag{12}$$

$$yy_4 = \frac{(y_2 + y_3 + y_4 + y_5 + y_6)}{5}, \tag{13}$$

Having received a working Kohonen compression network, you can start training a multilayer perceptron. The following architecture was chosen for the perceptron: 4 layers of neurons with an activation function – a logistic sigmoid, 30 neurons on the input layer, 40 neurons on the first hidden layer, 10 on the second hidden layer and 2 neurons on the output layer. The training takes place according to the following algorithm: a compressed image of the simulated mode is fed to the input, as a result of the network operation, two output neurons output values from 0 to 1. The difference between the output values of the network and the evaluation of the quality of the execution of the mode specified during the simulation is calculated. Next, the weights of the neural network are adjusted by the method of back propagation of the error. Training continues until the total quadratic error is less than some value specified by the operator. In this paper, this value was equal to 0.25.

Once again, we will fully form the algorithm of the recognition system. Initially, a record of the flight in which the test mode is performed is fed to the input of the system. A search window with a changing size moves throughout the flight record within the limits set in advance. Each image that enters the window is smoothed and interpolated into a vector with a dimension equal to 50 points. The resulting vector is fed to the input of the primary Kohonen recognition network. If any of the neurons of the control group defined at the training stage takes a value greater than 0.9, the time values corresponding to this section of the flight are recorded in the array. This creates a slicing of the entire mode into sections where a high-quality maneuver is presumably located. Further, these sections are encoded by the second Kohonen compression network and the outputs of this network are fed to the input of a multilayer perceptron, which determines the quality of the recognized mode. The evaluation of the mode varies from 0 to 1, where 1 is a qualitatively executed mode, 0 – this mode is not suitable for determining the characteristics of the aircraft.

Consider in detail the work of the primary Kohonen recognition network with sequential mode recording. Let a window capturing 100 points move according to the record shown in Fig. 12.

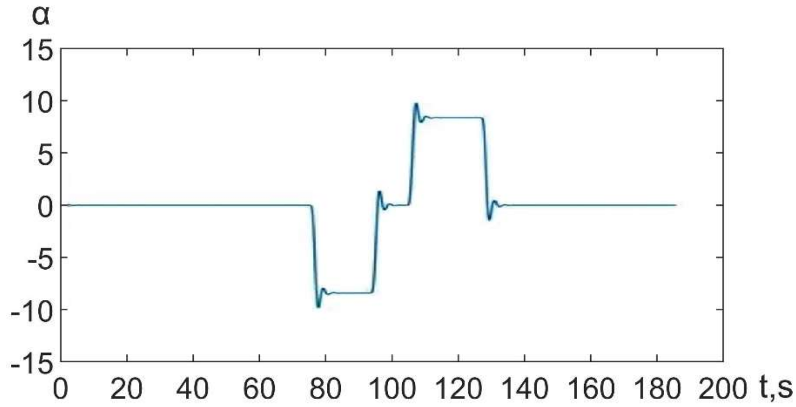



**Fig. 12.** An example of a flight recording of a change in the angle of attack

Based on the results of the training, it was determined that when the beginning and end of the angle of attack recording are applied during the test mode, the maximum values are taken by 2 and 10 neurons, respectively. Fig. 13 shows the time values of these neurons along the entire recording of the test maneuver.

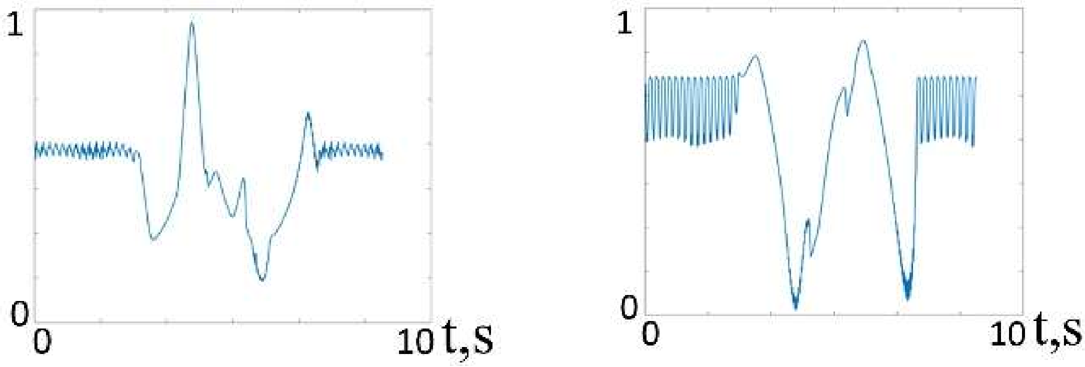



**Fig. 13.** Values of neurons along the recording of the test maneuver

As you can see, the neurons take a value greater than 0.9 in those parts of the recording where the beginning and end of the mode are located. The remaining neurons of the Kohonen network are excited similarly, when other flight parameters are applied.

The probability of recognition was estimated for noise with a maximum relative amplitude from 1% to 50% with different recording mode integrity - up to 60% of the duration, with a 20% cut off from the beginning and end of the recording. After conducting 1000 tests for each mode, the probability of recognition was estimated as the ratio of the number of correct recognitions to the total number of tests.

Fig. 14–15 below shows the dependences of the recognition probability on the noise amplitude according to the test results for various specified distortion parameters. The probability of recognition varies from 0 to 1.

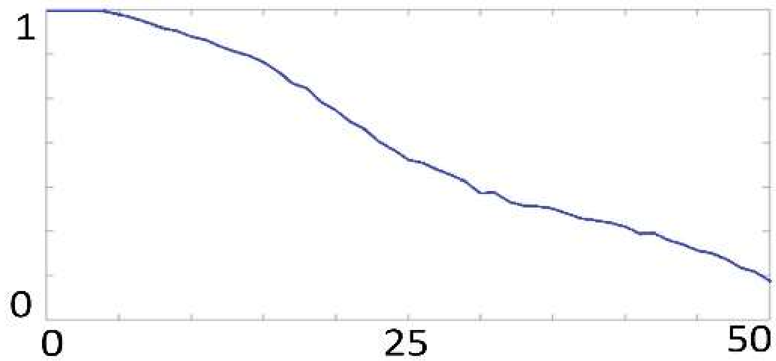



**Fig. 14.** Test result when adding noise from 0% to 50%

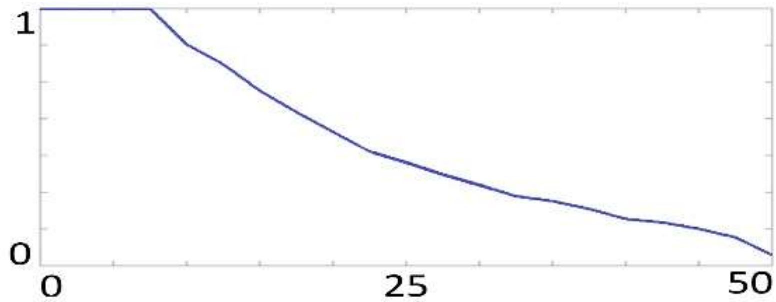



**Fig. 15.** Test result when cropping the image. The image was cropped to 20% at both ends.

It can be seen from the results that the algorithm used recognizes the modes with sufficient probability with large distortions. At the current time, the relative amplitude of the noise in the telemetry data is approximately 5%. With such a measurement error, the probability of recognizing the mode is more than 80%, even when only 60% of the recording is submitted. The probability of recognition is 50% with a maximum amplitude of relative error greater than 50%.

Now let's consider the complex operation of the recognition system with the sequential submission of an entire record of a simulated test flight, in order to verify the operation of the system in real time. Fig. 16 shows a graph of changes in the vertical overload of the aircraft during flight. This record was processed by the recognition system, and three modes were found satisfying the specified execution conditions. The fourth mode was not recognized, because the steering wheel was not correctly shifted on it.

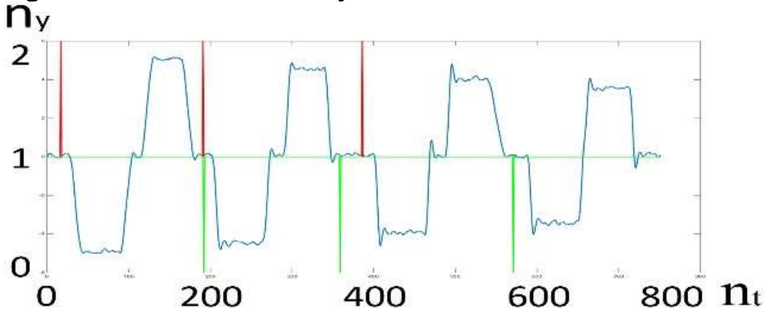



**Fig. 16.** Allocation of recognized modes by the processing system

Red strokes (see Fig. 16) the system marked the beginning of the recognized modes, and green marked their end. As you can see, all the qualitatively executed modes were recognized. The recognition system rated the first mode 0.42260, the second 0.9063, and the third 0.7609. When performing the first mode, it moved smoothly, which does not meet the quality requirements of the test mode, the second mode meets the quality requirements of the test mode. When performing the third mode at the end, the handle moved too smoothly. Fig. 17 shows a graph of the change in the angular velocity $w_z$.

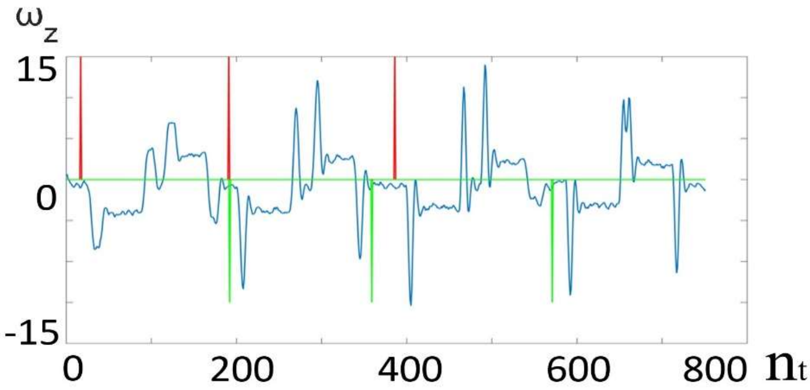



**Fig. 17.** Change in the angular velocity of the pitch

## 6. CONCLUSION

In this paper the major types of neural networks and methods of their training in reference modes were studied. Two types of networks were used: a multilayer perceptron and a Kohonen network. The multilayer perceptron in this work is used to obtain a final conclusion about the validity of the performed mode. The Kohonen network is used both to determine the intervals of the estimated finding of modes close to the reference, and to compress the input image for the perceptron.

The neural networks used in this work and trained on reference images showed excellent results in recognizing modes in the data obtained by the flight experiment. The described network learning algorithms can be used to work with images of other modes.

An algorithm has been developed for recognizing the flight experiment mode using the described networks on the example of data on the deviation of the elevator, the magnitude of the overload and the angle of attack when performing the "dacha pitch" maneuver. This algorithm can be used to recognize other flight modes.

The operation of the system has been tested when simulating real-time data acquisition. The results obtained show that the system recognizes all qualitatively executed modes and their assessment corresponds to reality. The system has shown sufficient stability of operation with a relative noise of up to 30%

REFERENCES

1. Kotik, M. G., Pavlov, A. V., Pashkovsky, I. M., Sardanovsky, Yu. S., & Shchitaev, N. G. (1965). *Letnye ispitaniya samoletov* [Flight tests of aircraft]. Russia, Moscow: Mashinostroenie, [in Russian].

2. Pashkovsky, I. M. (1975). *Ustoichivost' i upravlyaemost' samoleta* [Stability and controllability of the aircraft]. Moscow, Russia: Mashinostroenie, [in Russian].

3. Gorelik A. L., & Skripkin, V. A. (1977). *Metody raspoznavaniya* [Recognition methods]. Moscow, Russia: Vischaya shkola, [in Russian].

4. Aksenov, S. V., & Novoseltsev, V. B. (2006). *Organizaciya i ispolzovanie neironnyh setei (metody i tehnologii)* [Organization and use of neural networks (methods and technologies)]. Tomsk, Russia: Izdatelstvo NTL, [in Russian].

5. Haikin, S. (2006). *Neironnie seti: polniy kurs, 2-e izdanie* [Neural networks: a complete course, 2nd edition]. Moscow, Russia: I.D. Williams, [in Russian].