Parameter Debugging (Regulation) Method of Helicopters Aircraft Engines in Flight Modes Using Neural Networks

Serhii Vladov, Yurii Shmelov and Ruslan Yakovliev

Kremenchuk Flight College of Kharkiv National University of Internal Affairs, Peremohy street, 17/6, Kremenchuk, Poltava Region, Ukraine, 39605

Abstract

The work is devoted to solving the urgent scientific and practical problem of parameters debugging (regulation) of helicopters aircraft gas turbine engines (GTE) in flight modes using neural network technologies. A universal mathematical model has been developed of parameters debugging (regulation) aviation gas turbine engines, which is based on the operation algorithm of the control device, which leads to the elimination of the mismatch of the control elements of aircraft GTE, using the A. M. Lyapunov functions. To solve this problem, a neural network of feedforward propagation was used in the work with the use of adaptive elements, which made it possible to obtain a graph of the dependence of the objective function of specific fuel consumption on the r.p.m. and, as a consequence, to clarify the values of these parameters.

Keywords

Aircraft engine, neural network, debugging (regulation), scattering ellipse, specific fuel consumption, r.p.m., adaptive elements, training.

1. Introduction

An analysis of helicopter accidents by engine type showed that helicopter accidents with reciprocating engines were more frequent in the early stages of helicopter operation, but as gas turbine engines became more popular, the trend changed significantly close to 10 years ago. Recent data show that the number of emergency and catastrophic situations of helicopters with gas turbine engines has increased compared to the number of accidents and disasters of helicopters with piston engines. As a result, further research is needed to improve the safety of flights of helicopters with gas turbine engines. Thus, the task of debugging (regulating) the parameters of aircraft gas turbine engines of helicopters, solved in this work, is today an urgent scientific and practical task [1–4].

2. Literature review

An analysis of works in this area shows [5–7] that the currently existing algorithms and programs that implement this process are not without drawbacks, among which the main ones are:

– the lack of a universal methodology that implements this task (most enterprises in the industry are guided by their own developments);

– the requirement for the availability of large volumes of a priori and a posteriori information on the fleet of gas turbine engines;

– assignment of tight tolerances for each debugged parameter;

– significant time spent on the process of debugging the parameters of the gas turbine engine associated with the need to solve the optimization problem: minimization of the quality / time functional, etc.

Information Technology and Implementation (IT&I-2021), December 01–03, 2021, Kyiv, Ukraine

EMAIL: ser26101968@gmail.com (S. Vladov); nviddil.klk@gmail.com (Yu. Shmelov); ateu.nv.klk@gmail.com (R. Yakovliev)
ORCID: 0000-0001-8009-5254 (S. Vladov); 0000-0002-3942-2003 (Yu. Shmelov); 0000-0002-3788-2583 (R. Yakovliev)

– In order to eliminate these shortcomings, in this work, a method is developed for solving the problem of debugging the parameters of a gas turbine engine, based on the use of neural network technologies. The peculiarity of the formulation and solution of this problem is that when constructing a neural network, as before, only experimentally obtained information is used, while the classical methods for solving this problem [5–7] require the use of average mathematical models of gas turbine engines, a description of the physics of the flowing processes, etc.

3. Problem statement of debugging the parameters of helicopters aircraft engines

According to [8], it is assumed that the parameters of an adjusted, normally functioning GTE in the space of controlled parameters, for example, on the plane of thermos gas dynamic parameters X_1^* and X_2^* (fig. 1), correspond to a given nominal engine operating mode: $X_1^* = X_{1opt}^*$, $X_2^* = X_{2opt}^*$. It is assumed that the characteristics of the fleet of serviceable engines for the same measured parameters give some scatter relative to the specified nominal point, forming an scattering ellipse (in multidimensional space – an ellipsoid) (fig. 1).

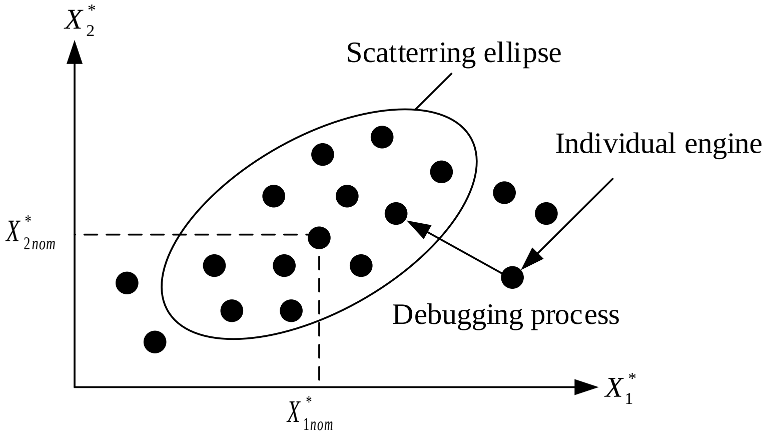


Figure 1: Scattering ellipse of the parameters of aircraft GTE [8, 9]

The departure of the operating point beyond this ellipse corresponds to abnormal changes in the parameters of an individual engine. Then the purpose of debugging the GTE parameters is to return the "dropped out" point to the ellipse (ellipsoid) by smoothly adjusting the GTE structural elements. According to [8], the controlled element of the GTE design is the throat area of the nozzle apparatus of the F_{noz} engine. This parameter is available only in turbojet and turbofan engines and is absent in turboshaft engines (GTE with a free turbine), which are used as part of helicopter power plants.

The solution to this problem in a neural network basis is presented in the form of the following sequence of steps [8]:

– formation of a training sample based on the test results of the GTE fleet;

– determination of the boundaries of variation of the variable parameter;

– construction of a neural network model of an average gas turbine engine, the input parameters of which are the size of the jet nozzle diameter, and the outputs are the thermos gas dynamic parameters of the engine;

– building an adjustment curve;

– calculation of the required correction of the diameter of the jet nozzle of an individual GTE;

– values clarification of thermos gas dynamic parameters of the adjusted engine for the corrected value of the diameter of the jet nozzle.

In [9], an attempt was made to apply the solution to the problem of debugging the parameters of a gas turbine engine in a neural network basis [8] when operating the TV3-117 aircraft engine. It belongs to the class of aircraft gas turbine engines with a free turbine and it is used as part of the power plant of the Mi-8MTV helicopter.

4. Development of a universal mathematical model for debugging the parameters of aircraft GTE

A universal mathematical model for debugging the parameters of an aircraft GTE is based on the operation algorithm of the control device, leading to the elimination of the mismatch ε, is calculated for each control element (CR) of the aviation GTE. The most universal is the approach based on the application of A. M. Lyapunov functions [10, 11]:

$$V_1 = F\left(\varepsilon,\dot{\varepsilon},u_n,\alpha_1\right);\; V_2 = F\left(\varepsilon,\dot{\varepsilon},u_p,\alpha_2\right);\dots\; V_i = F\left(\varepsilon,\dot{\varepsilon},u_i,\alpha_i\right); \tag{1}$$

where α_i – parameter characterizing the position of the CR. In this case, the time derivative of each A.M. Lyapunov function and the non-positiveness condition $\frac{dV_i}{dt} \le 0$ is imposed on it. Then, based on the stability conditions of the tuning process, the tuning algorithm is determined in the form $\dot{\alpha}_i = F\left(\varepsilon,\dot{\varepsilon},u_n,u_p\right)$. Thus, the debugging of the aircraft GTE regulator is carried out according to the given parameters u_n and u_p.

When using the direct method of A.M. Lyapunov, according to the stability conditions, the mutual influence of the CE position on the output parameters of the fuel regulator is compensated. Since the tuning algorithm for each CE will be convergent, then in the case of simultaneous tuning of the agreed characteristics of all CE, the entire process of debugging the fuel regulator will be convergent and stable. The main disadvantage of the proposed method is the possible inconsistency of the characteristics of various CE. Let us consider the process of debugging the uncoordinated characteristics of helicopters aircraft gas turbine engines using the example of a fuel dispenser (fig. 2).

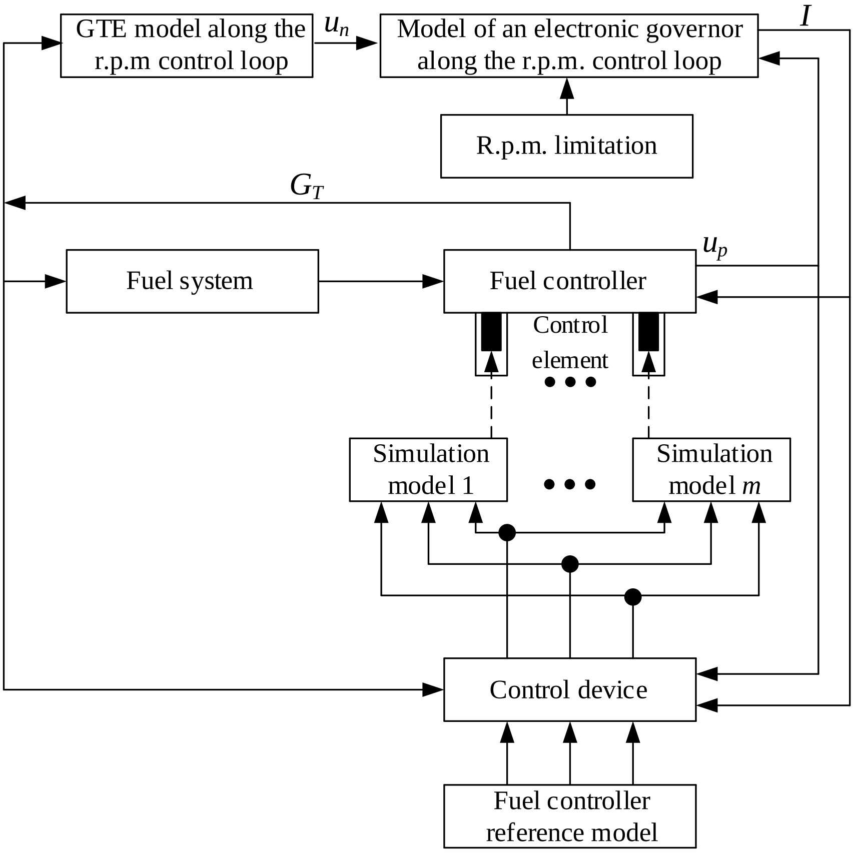

GTE model along the r.p.m control loop
u_n
Model of an electronic governor along the r.p.m. control loop
I
R.p.m. limitation
G_T
u_p
Fuel system
Fuel controller
Control element
Simulation model 1
Simulation model *m*
Control device
Fuel controller reference model

Figure 2: Debugging diagram of an aircraft gas turbine engine fuel dispenser using an optimization algorithm

The debugging of the fuel dispenser is carried out according to the reference model of the dispenser, which consists of series-connected aperiodic links with transfer functions:

$$W_1^E\left(s\right)=\frac{k_1^E}{T_1^E s+1};\;\; W_2^E\left(s\right)=\frac{k_2^E}{T_2^E s+1};\;\; W^E\left(s\right)=W_1^E\left(s\right)\cdot W_2^E\left(s\right). \tag{2}$$

In the first link (electromechanical actuator), the current signal *I* is converted into an angle of rotation *α* or displacement of the output element of the actuator; in the second link (metering unit), the angle of rotation *α* or displacement of the output element of the actuator is converted into a fuel consumption G_T.

The reference model of a fuel dispenser for helicopters aircraft GTE is linear:

$$\dot{\alpha}=A^E\cdot\alpha+B^E\cdot I;\;\; \dot{G}_T=C^E\cdot G_T+D^E\cdot\alpha; \tag{3}$$

where A^E, B^E, C^E, D^E – specified reference coefficients of differential equations, which are calculated according to the expressions:

$$A^E=\frac{-1}{T_1^E};\;\; B^E=\frac{-k_1^E}{T_1^E};\;\; C^E=\frac{-1}{T_2^E};\;\; D^E=\frac{-k_2^E}{T_2^E}. \tag{4}$$

A real fuel dispenser for an aircraft GTE is a nonlinear continuous object, i.e.:

$$\dot{\alpha}=A\cdot\Psi\left(\alpha\right)+B\cdot\Phi\left(I\right);\;\; \dot{G}_T=C\cdot Q\left(G_T\right)+D\cdot U\left(\alpha\right); \tag{5}$$

where *A*, *B*, *C*, *D* – coefficient matrices; $\Psi(\alpha)$, $\Phi(I)$, $Q(G_T)$; $U(\alpha)$ – nonlinear functions.

The equation of the adaptive model of the identified parameters has the form [12, 13]:

$$\dot{\alpha}^M=A^M\cdot\Psi\left(\alpha^M\right)+B^M\cdot\Phi\left(I\right);\;\; \dot{G}_T^M=C^M\cdot Q\left(G_T^M\right)+D_M\cdot U\left(\alpha^M\right); \tag{6}$$

where A^M, B^M, C^M, D^M – tunable coefficients equal at the end of the identification process to the coefficients of the equations describing the fuel dispenser; $\dot{\alpha}^M$, α^M, $\dot{G}_T^M$, G_T^M – output parameters of the adaptive model.

When subtracting from equations (6) equations (5):

$$\Delta\dot{\alpha}=\Delta A\cdot\Psi\left(\alpha\right)+A^M\cdot\Delta\Psi+\Delta B\cdot\Phi\left(I\right)+B_M\cdot\Delta\Phi; \tag{7}$$

$$\Delta\dot{G}_T^M=\Delta C\cdot Q\left(G_T^M\right)+C^M\cdot\Delta Q+\Delta D_M\cdot U\left(\alpha^M\right)+D_M\cdot\Delta U; \tag{8}$$

where $\Delta A=A-A^M$; $\Delta B=B-B^M$; $\Delta C=C-C^M$; $\Delta D=D-D^M$; $\Delta\alpha=\alpha-\alpha^M$; $\Delta\dot{\alpha}=\dot{\alpha}-\dot{\alpha}^M$; $\Delta G_T=G_T-G_T^M$; $\Delta\dot{G}_T=\dot{G}_T-\dot{G}_T^M$; $\Delta\Psi\left(\alpha\right)=\Psi\left(\alpha\right)-\Psi\left(\alpha^M\right)$; $\Delta Q=Q\left(G_T\right)-Q\left(G_T^M\right)$; $\Delta U\left(\alpha\right)=U\left(\alpha\right)-U\left(\alpha^M\right)$.

The residual signals are equal:

$$\varepsilon_1=\Delta\dot{\alpha}-A^M\cdot\Delta\Psi=\Delta\alpha\cdot\Delta\Psi+\Delta B\cdot\Phi\left(I\right); \tag{9}$$

$$\varepsilon_2=\Delta\dot{G}_T-C^M\cdot\Delta Q-D^M\cdot\Delta U=\Delta C\cdot Q\left(G_T\right)+\Delta D\cdot U\left(\alpha\right). \tag{10}$$

The values $\Delta\dot{\alpha}$, *α*, α^M, $A^M\cdot\Delta\Psi$, $\Delta\dot{G}_T$, G_T, G_T^M, $C^M\cdot\Delta Q$, $D^M\cdot\Delta U$ are directly observable or calculated through directly measurable quantities.

To minimize the residual vectors, the Lyapunov functions were chosen in the form of positive definite quadratic forms:

$$V_1=\frac{1}{2}\left(\Delta A\cdot K\cdot\Delta A^T+\Delta B\cdot L\cdot\Delta B^T\right);\;\; V_2=\frac{1}{2}\left(\Delta C\cdot M\cdot\Delta C^T+\Delta D\cdot N\cdot\Delta D^T\right); \tag{11}$$

where *K*, *L*, *M*, *N* – positive definite diagonal matrices of given constant coefficients; ΔA^T, ΔB^T, ΔC^T, ΔD^T – transposed matrices of differences of coefficients.

Derivatives of quadratic forms are:

$$\dot{V}_1=\Delta A\cdot K\cdot\Delta\dot{A}^T+\Delta B\cdot L\cdot\Delta\dot{B}^T;\;\; \dot{V}_2=\Delta C\cdot M\cdot\Delta\dot{C}^T+\Delta D\cdot N\cdot\Delta\dot{D}^T; \tag{12}$$

provided

$$\Delta\dot{A}^T=\frac{-\varepsilon_1\cdot\alpha}{K}=\frac{-\left(\Delta A\cdot\Psi\left(\alpha\right)+\Delta B\cdot\Phi\left(I\right)\right)\cdot\alpha}{K}; \tag{13}$$

$$\Delta\dot{B}^T=\frac{-\varepsilon_1\cdot I}{L}=\frac{-\left(\Delta A\cdot\Psi\left(\alpha\right)+\Delta B\cdot\Phi\left(I\right)\right)\cdot I}{L}; \tag{14}$$

$$\Delta\dot{C}^T = \frac{-\varepsilon_2 \cdot G_T}{M} = \frac{-\left(\Delta C \cdot Q(G_T) + \Delta D \cdot U(\alpha)\right) \cdot G_T}{M}; \tag{15}$$

$$\Delta\dot{D}^T = \frac{-\varepsilon_2 \cdot \alpha}{N} = \frac{-\left(\Delta C \cdot Q(G_T) + \Delta D \cdot U(\alpha)\right) \cdot \alpha}{N}; \tag{16}$$

$$\dot{V}_1 = \left(\Delta A \cdot \Psi(\alpha) + \Delta B \cdot \Phi(I)\right)^2; \ \dot{V}_2 = \left(\Delta C \cdot Q(G_T) + \Delta D \cdot U(\alpha)\right)^2; \tag{17}$$

process is steadily converging.

Debugging equations implemented in analyzers can be written in matrix form:

$$\dot{A}^M = \varepsilon_1 \cdot \left|K \cdot \Psi(\alpha)\right|^T \text{ or } A^M = \int \varepsilon_1 \cdot \left|K \cdot \Psi(\alpha)\right|^T dt; \tag{18}$$

$$\dot{B}^M = \varepsilon_1 \cdot \left|L \cdot \Phi(I)\right|^T \text{ or } A^M = \int \varepsilon_1 \cdot \left|L \cdot \Phi(I)\right|^T dt; \tag{19}$$

$$\dot{C}^M = \varepsilon_2 \cdot \left|M \cdot Q(G_T)\right|^T \text{ or } C^M = \int \varepsilon_2 \cdot \left|M \cdot Q(G_T)\right|^T dt; \tag{20}$$

$$\dot{D}^M = \varepsilon_2 \cdot \left|N \cdot U(\alpha)\right|^T \text{ or } D^M = \int \varepsilon_2 \cdot \left|N \cdot U(\alpha)\right|^T dt. \tag{21}$$

The identified values of the A^M, B^M, C^M, D^M coefficients describing a real fuel meter are compared with the A^E, B^E, C^E, D^E values of the reference meter model. Signals of differences of identifiable and reference coefficients

$$\delta A = A^M - A^E; \ \delta B = B^M - B^E; \ \delta C = C^M - C^E; \ \delta D = D^M - D^E \tag{22}$$

used for debugging the fuel dispenser. The amount of movement of the actuators is determined by the sensitivity of the fuel meter to the movement of the CR.

5. Development of a neural network classifier using a feedforward neural network

The neural network stores the information portrait of the average statistical engine in one of its operating modes and, when the value *n* is fed to its input, calculates the values of the reduced parameter G_T using a universal mathematical model for debugging the parameters of aircraft GTE. The architecture of the neural network is a three-layer feedforward network [8, 9] using adaptive elements (fig. 3).

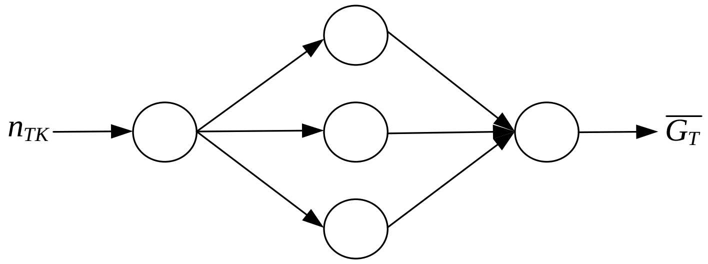


Figure 3: Neural network structure

In general, some adaptive element transforms **T** the input vector **x** into the output vector **y**:

$$\mathbf{y} = \mathbf{T}(\mathbf{x}, \boldsymbol{\theta}); \tag{23}$$

where **θ** – vector of the parameters of this transformation. Each adaptive element has a certain vector of parameters, the value of which is determined in the learning process. The adaptive elements are interconnected in the network structure with the help of bidirectional channels that ensure the passage of signals in the forward and reverse directions, which is schematically shown in fig. 4 [14].

It is assumed that there are two main operating modes for a neural network: operation mode and training mode. In the operating mode, direct passage is used, which allows, with a known transformation form **T** and a vector of parameters **θ**, to obtain the response $\mathbf{y}_F$ of the element to some input signal $\mathbf{x}_F$. In teach mode, the parameter vector is adjusted based on the error signal $\mathbf{x}_B$.

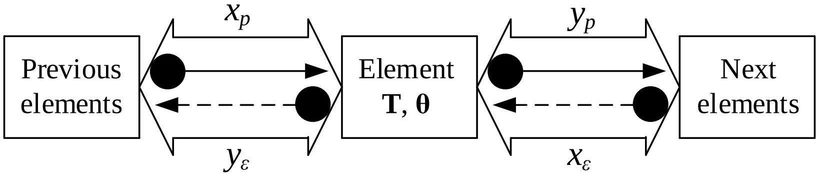


Figure 4: Structure of signaling connections of the adaptive element

6. Neural networks training

In this paper, we consider supervised learning [15], in which a set of examples from a training set is consistently presented to a trained network. Examples are pairs of reference inputs and desired outputs. The learning process takes place cyclically, at each iteration, the signals are calculated for forward and backward propagation, after which the error signals are used to form local gradients of the vectors of the adaptable parameters. The calculated local gradients are used for the subsequent adjustment of the adapted parameters [16, 17]. The training modes used are the sequential mode (online), in which the parameters are adjusted after each example, and the batch, in which the adjustment is based on the cumulative local gradient – the sum of local gradients over all iterations of the examples from the training set. In both modes, the full cycle of presentation of a set of training patterns, which ends with the adjustment of parameters, is called the network training epoch. To quantitatively assess the quality of the network, a loss function is introduced that has the meaning of the root mean square error (RMSE):

$$E = \frac{1}{2N}\sum_{n=1}^{N}\left(t_n - z_n\right)^2; \tag{24}$$

where N – number of templates presented; t_n – desired, or target, output signal of the network in the n-th pattern; z_n – output signal calculated by the network when the input signal is supplied from the n-th template.

At this stage of the research, the authors consider only autonomous learning methods that use only those signals that are present in the considered element to adjust the vector of parameters of an adaptive element. The corrective change $\Delta\theta_m$ for the m-th element θ_m from the vector of adaptable parameters $\boldsymbol{\theta}$ is calculated based on information obtained only from its local gradient $\delta\theta_m$ and the history of changes in this local gradient over epochs. Gradient descent method is the simplest method for training a network. The adaptable parameters θ is corrected by the value $\Delta\theta_i$ according to the expression:

$$\theta_{i+1} = \theta_i + \Delta\theta_i; \tag{25}$$

where θ_{i+1} – corrected value of the parameter, θ_i – original value of the parameter, $\Delta\theta_i$ – corrective change. In this case, the magnitude of the corrective change is determined by the expression:

$$\Delta\theta_i = -\varepsilon \cdot \delta\theta_i; \tag{26}$$

where ε – coefficient of the learning rate, $\delta\theta_i$ – local gradient of the element.

In expression (26), the minus sign in front of the coefficient is necessary to change the parameter θ as an argument of the function E in the direction of decreasing the value of the latter. It is important to pay special attention to the choice of the learning rate coefficient ε: a small value of the coefficient will lead to an increase in the time (number of iterations) required for training, but too large. A value will lead to destabilization of training due to an excessive increase in the parameter so that it will "slip" the optimal value. The value of the coefficient ε is taken from the condition:

$$0 < \varepsilon < 1. \tag{27}$$

The gradient descent method [18] is the basic method on the basis of which other autonomous methods of the first and so-called quasi-second order are built.

One of the obvious improvements of the gradient descent method is the addition of the effect of inertia (momentum) when changing parameters. It is a method of averaging, which in some cases makes it possible to significantly increase the stability of the learning process (the parameters reach their optimal values θ_{opt}). This method, in general, uses the average value of parameter changes in previous epochs to calculate the parameter measurement in the current epoch, which makes the parameter change smoother. The exponential average of the parameter measurements for all previous epochs is used; in this case, the expression for calculating the correction (26) takes the form:

$$\Delta\theta_i = \mu \cdot \Delta\theta_{i-1} - (1-\mu)\cdot \delta\theta_i; \tag{28}$$

where $\Delta\theta_i$ – correcting change in the current (i-th) epoch, $\Delta\theta_{i-1}$ – correcting change in the previous epoch, μ – coefficient of inertia, determines the measure of the influence of previous adjustments on the current one and, as a rule, is selected based on the condition $0 < \mu < 1$, ε – learning rate coefficient, $\delta\theta_i$ – local gradient of the element.

The left side in expression (28) represents the influence of the previous adjustments of the value of the parameter θ on the current adjustment, while the corrective change of the previous epoch $\Delta\theta_{i-1}$ is weighted by the inertia coefficient μ, therefore, the larger the coefficient of inertia, the stronger influence of the history of the parameter change on the current change. The right-hand side of the expression repeats the corresponding expression (26) for the gradient descent method, but includes a weighting coefficient $(1 - \mu)$ to take into account the share of the influence of previous epochs. In the case when the coefficient of inertia μ is equal to zero, the momentum method degenerates into a gradient descent method, while the history of measuring correction values for previous epochs does not affect the correction value of the current epoch (expression (28) formally turns into (26)).

In [19], the method known as Delta-Bar-Delta is considered in detail. In contrast to the gradient descent and moment method, the fundamental extension of this method is that an individual learning rate coefficient is introduced for each adaptive parameter. After each training epoch, both the adjustment of the adaptable parameters and the adjustment of the learning rate coefficient take place. To simplify the calculations, consider one of the adaptable parameters, which we denote by θ. To correct its rate of change, an auxiliary parameter f is introduced, which also changes with the course of the epoch number according to the rule:

$$f_i = \gamma \cdot f_{i-1} + (1-\gamma)\cdot \delta\theta_i; \tag{29}$$

where the coefficient γ ($0 < \gamma < 1$) determines the "memory depth" of the accumulation of the history of the previous values of the gradient. For the current epoch, the auxiliary parameter f defines the gradient accumulated over the previous epochs. If the sign of the gradient value of the current epoch $\delta\theta_i$ coincides with the sign of the coefficient f_i, then the learning rate increases, otherwise it decreases. The magnitudes of changes in the learning rate are determined by the expression:

$$\varepsilon_i = \begin{cases} \varepsilon_{i-1} + k, \text{ if } \delta\theta_i \cdot f_i > 0 \\ \varepsilon_{i-1} \cdot \varphi, \text{ if } \delta\theta_i \cdot f_i > 0 \end{cases} \tag{30}$$

The parameters φ and k, selected in the range from zero to one, determine how large the change in the training rate will be with each adjustment. The vector composed of the values of the velocities ε_i calculated by the expression (30) and the value of the gradient $\delta\theta_i$ are used in the expression (26) to calculate the new value of the vector of the adaptable parameters.

7. Application of adaptive elements in a neural network classifier

The use of autonomous teaching methods makes it possible to build relatively simple models of elements, the simplest of which are shown in fig. 5, *a – d*. Among the adaptive elements, the amplifier occupies a central place, since it is in it that the network coefficients are adjusted in accordance with the selected training method. The amplifier is shown schematically in fig. 5, *a*, it has one input and one output, while the output signal is determined by the expression:

$$y_f = w \cdot x_f. \tag{31}$$

When the signal propagates back, the amplifier does not change its behavior, i.e., the output signal is w times the amplified input signal:

$$y_b = w \cdot x_b; \tag{32}$$

where x_b – back propagating input signal; y_b – back propagating output signal; w – gain, which is given by:

$$w = f'\left(x_f\right); \tag{33}$$

which shows the dependence of the behavior of the element during backpropagation on the value of the first derivative of the function *f* at the point x_f.

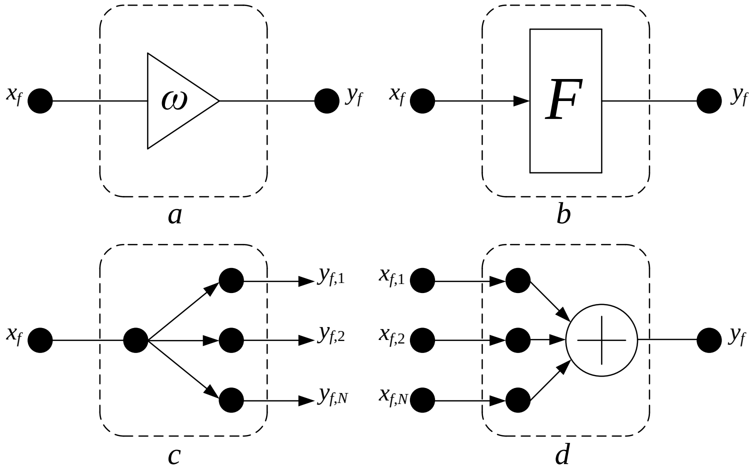


Figure 5: The simplest elements of a neural network: *a* – adaptive multiplier; *b* – inertialess functional converter; *c* – splitter; *d* – adder [14]

The amplifier in the process of training a neural network is able to change its own gain, while realizing the adaptive properties of the network. A functional converter is generally an element with one input, one output and a known transfer function *f*. The behavior of all functional transducers during forward and backward passage of signals is determined by their functions *f* and first derivatives f'. The functional converter is shown in fig. 5, *b*, its output signal for forward propagation is determined by the expression:

$$y_f = f\left(x_f\right); \tag{34}$$

where x_f – forward-propagated input; y_f – forward-propagated output; *f* – transform function. In backpropagation, the functional transducer is an amplifier, and the output signal is given by expression (32).

Functional transducers are part of neurons, forming various types of the latter, while the transformation function is a function of neuron activation. For the presented elements, signal conversion functions were obtained for forward signal propagation and back propagation of the error. The composition of the simplest elements makes it possible to build classical structural elements of a feedforward network – a neuron and a neural layer, which, in turn, can also be described using the notation of adaptive elements. The classical neuron of the McCulloch-Pitts model in the paradigm of constructing a neural network based on simple adaptive elements [20] can be implemented algorithmically, or by a composition of the simplest elements (adders, amplifiers and functional converters). The algorithmic implementation, undoubtedly, has better performance, however, for greater clarity, the compositional model will be considered, it is shown in fig. 6, *a*. Input amplifiers are individual for each input signal, so their coefficients are, in fact, synoptic weights w_i of the neuron. The functional transformer at the output as a transformation function contains the activation function of the neuron *f*. The block diagram of a neuron in reverse propagation is shown in fig. 6, *b*.

The output signal of the neuron is formed as a result of the passage of the sum of the weighted input signals through the functional transducer:

$$y_f = f\left(\sum_{i=1}^{N} w_i \cdot x_{f_i}\right); \tag{35}$$

where y_f – output signal during direct transmission; *f* – activation function; x_{f_i} – *i*-th input signal; w_i – *i*-th synaptic weight.

When a signal propagates backward, the behavior of a neuron is determined by the behavior of its constituent elements, and in the general case, the output signal is determined by the expression:

$$y_{b_i} = x_b \cdot g \cdot w; \tag{36}$$

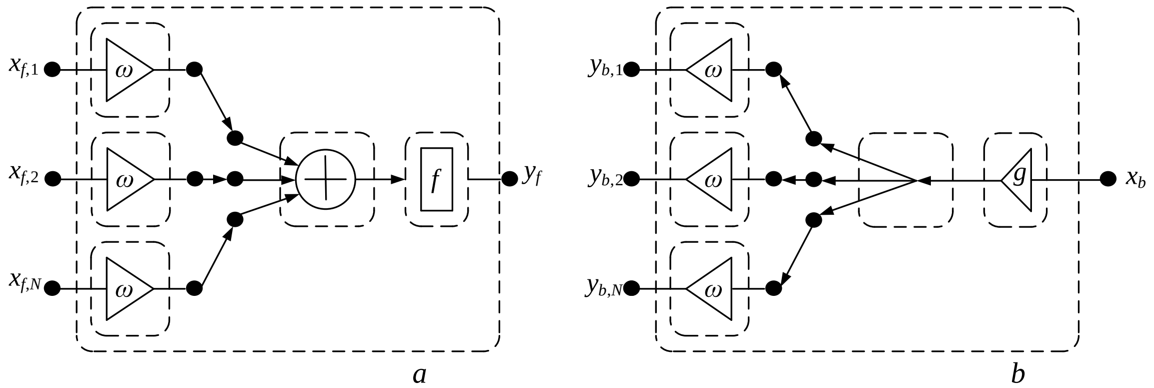


Figure 6: Neuron decomposition: *a* – with direct signal transmission; *b* – with back propagation of the error [14]

where y_{b_i} – *i*-th output signal of the network during backpropagation; x_b – input signal of the network during backpropagation; g – gain of the functional transducer during backpropagation; w_i – *i*-th synaptic weight. Taking into account the behavior of the functionally transformer (35), expression (36) takes the form:

$$y_{b_i} = x_b \cdot f'\left(\sum_{i=1}^{N} w_i \cdot x_{f_i}\right) \cdot w_i. \tag{37}$$

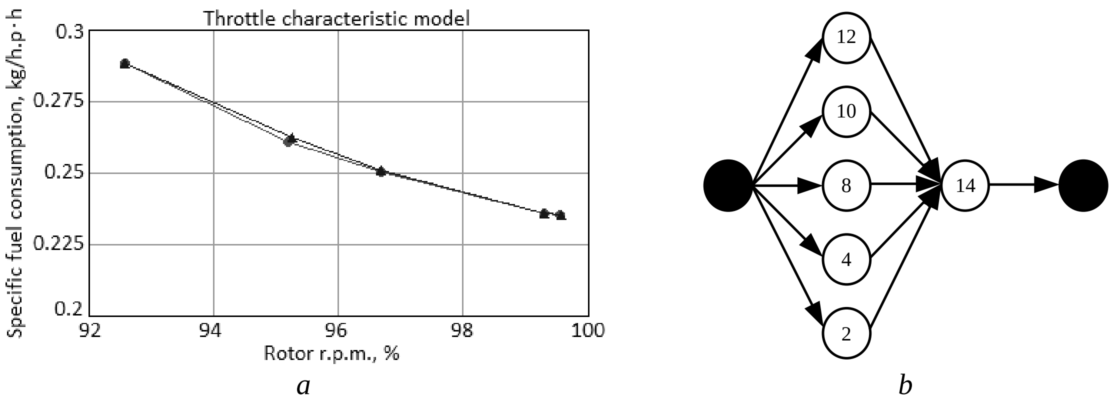
8. Results and discussion

To implement a software prototype using the universal modeling language (UML) [21], a class hierarchy was developed, Python was used as a programming language, which, on the one hand, provides a simple form of writing mathematical expressions, on the other hand, broad opportunities in the field of object-oriented programming. The software implementation is divided into two parts – basic and additional. The basic part is made as a package and includes the logic necessary for the application to work, but does not contain data input and output tools. The additional part is designed to work in an interactive mode as part of the Sage computer mathematics system, a software package with a free license, united by a single user and software interface. An additional part uses the capabilities of Sage for data input and output: generation of training sequences, displaying graphs, building graphs and tables.

The developed software prototype was used for numerical simulation of the application of a neural network to solve the problem of approximation and classification of input data.

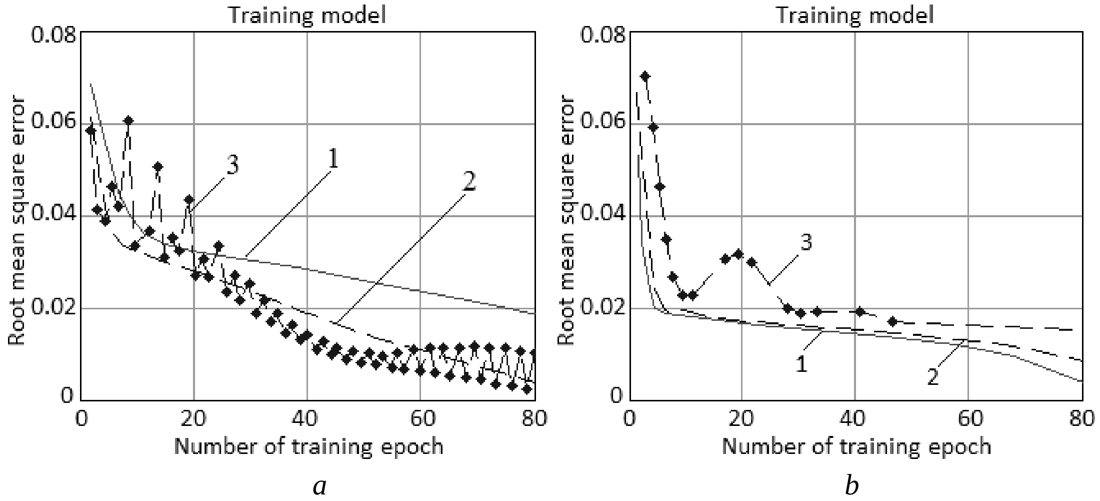
The dependence of the specific fuel consumption C_e of the TV3-117 aircraft engine on the rotational speed of the gas generator rotor r.p.m n_{TK} (element of the throttle characteristic of the engine) was used as an input signal in the approximation problem. The input data is shown in fig. 7, *a*, but by points that are approximated by broken lines for ease of perception. Depending on the training method used, for a certain number of iterations, the synaptic weights of the neural network approach the optimal values. In fig. 7, *a*, and the triangles show the received output signal of the network, approximated by a broken line together with the original signal. In fig. 7, *b* shows a neural network graph built directly using the tools of the developed software prototype.

The dependence of the root mean square error (RMSE) [22] during training for the gradient descent method is shown in fig. 8, *a*, where an increase in the initial coefficient of the training rate ε from 0.6 (continuous line) to 1.2 (dash-dotted line) allows to slightly speed up the training process and achieve better results in less time. However, a further increase in the training rate factor leads to a too fast adjustment of the synaptic weights. This leads to the fact that the weights oscillate around the optimal values, without reaching them – the RMSE takes on a similar oscillatory character. To demonstrate the effect of inertia, the learning rate coefficient ε = 1.2 was chosen, then the network was trained at three different inertia coefficients μ. The results of the influence of the effect of inertia are shown in fig. 8, *b*. An increase in the inertia coefficient in a number of cases leads to a slight change in the learning rate, and a further increase leads to a negative effect – the destabilization of the training process of the network as a whole.



a *b*

Figure 7: Input data: *a* – dependence $C_e = f(n_{TK})$ and the result of approximation; *b* – configuration of a neural network built by a software prototype



a *b*

Figure 8: Graph of the change in the mean square error during training for the methods: *a* – gradient descent (1 – ε = 0.6, μ = 0; 2 – ε = 1.2, μ = 0; 3 – ε = 1.8, μ = 0); *b* – gradient descent with inertia (1 – ε = 1.2, μ = 0.3; 2 – ε = 1.2, μ = 0.6; 3 – ε = 1.2, μ = 0.9)

The greatest interest is the Delta-Bar-Delta method [23], the results of which are shown in fig. 9, *a*. In [24], it was suggested that in most cases the optimal set of parameters for this method are the following values of the parameters of expressions (29), (30): γ = 0.3, φ = 0.7, k = 0.5, which in fig. 9 corresponds to a continuous line. As can be seen from this graph, the deviation of the parameters from the optimal values can, in a particular case, lead to both positive and negative results. In the case of an excessive decrease in the learning rate ratio (dashed line), there is a significant learning gap. A decrease in the coefficient γ, which determines the degree of influence of the error gradients obtained in previous epochs and, at the same time, an increase in the coefficient k, and, as a consequence, an acceleration in the growth of the rate coefficient, makes it possible to obtain a significant improvement in the quality of training.

To solve the problem of debugging the parameters of aircraft gas turbine engines of helicopters (on the example of TV3-117 aircraft gas turbine engine), as a training sample. We will use the values of the gas generator rotor r.p.m at the takeoff mode, reduced to absolute values [25, 26], given in table 1, and the parameters of the average engine fleet the next: $\overline{n_{TK}} = 0.994$, $\overline{C_e} = 0.977$.

Table 1
Fragment of the training set

n_{TK}	0.998	0.998	0.992	0.992	0.991	0.995	0.991	0.996	0.998	0.989	0.991	0.993
C_e	0.972	0.978	0.964	0.984	0.998	0.979	0.970	0.990	0.965	0.990	0.967	0.964

The graph of the change in the learning error of the neural network depending on the number of neurons in the hidden layer is shown in fig. 10, whence it follows that the training error of the neural network is minimal when the number of neurons in the hidden layer is equal to 3 [8, 9].

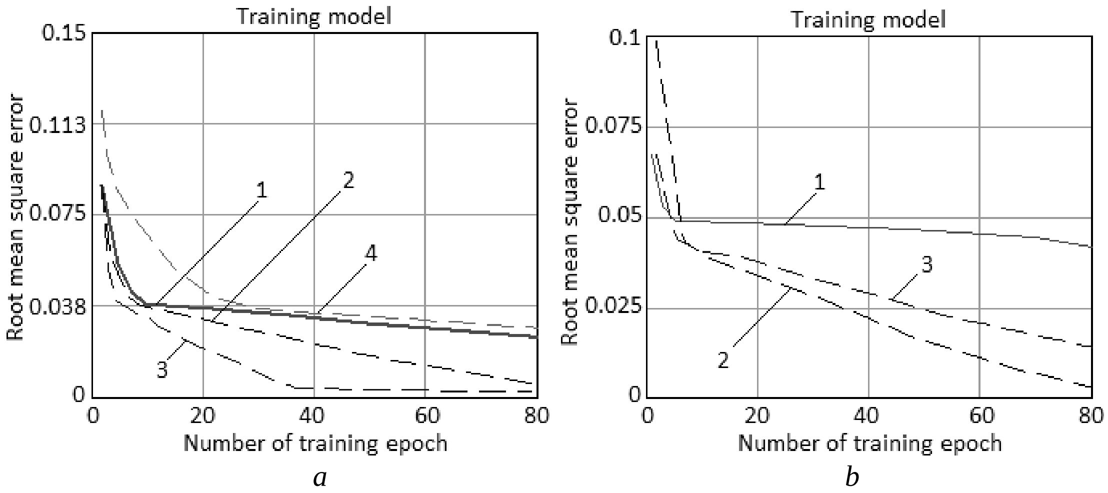


Figure 9: Graph of the change in the mean square error during training: *a* – for the Delta-Bar-Delta method (1 – $\varepsilon = 1.2$, $\mu = 0$, $\gamma = 0.7$, $\varphi = 0.5$, $k = 0.01$; 2 – $\varepsilon = 1.2$, $\mu = 0$, $\gamma = 0.7$, $\varphi = 0.1$, $k = 0.01$; 3 – $\varepsilon = 1.2$, $\mu = 0$, $\gamma = 0.3$, $\varphi = 0.5$, $k = 0.7$; 4 – $\varepsilon = 1.2$, $\mu = 0$, $\gamma = 0.7$, $\varphi = 0.1$, $k = 0.01$); *b* – for all presented methods (1 – $\varepsilon = 1.2$, $\mu = 0.3$; 2 – $\varepsilon = 1.2$, $\mu = 0$; 3 – $\varepsilon = 1.2$, $\mu = 0$, $\gamma = 0.7$, $\varphi = 0.5$, $k = 0.1$).

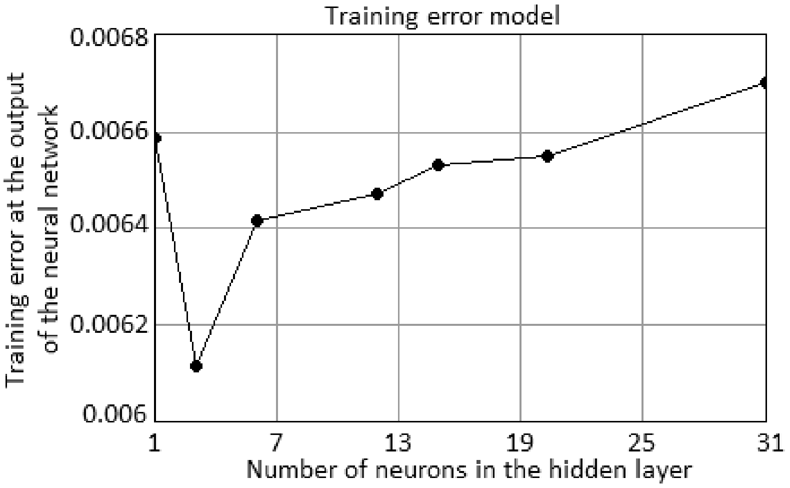


Figure 10: Graph of the dependence of the neural network learning error on the number of neurons in the hidden layer [8, 9]

Adjustment curve $C_e = f\left(\overline{n_{TK}}\right)$ [8, 9] according to fig. 7, *a* is represented as:

$$C_e(n) = 0.0016 \cdot n_{TK}^4 - 0.0195 \cdot n_{TK}^3 + 0.0864 \cdot n_{TK}^2 - 0.1774 \cdot n_{TK} + 0.4083. \qquad (38)$$

where $\overline{n_{TK}} = \dfrac{n_{TK}}{n_{TK\max}}$ – relative value of the gas generator rotor r.p.m.

Fig. 11 shows a graph of dependence of the objective function $C_e(n) \to \min$ [8, 9] from the value of the gas generator rotor r.p.m n_{TK}. In this case objective function minimum 0.20 is reached at the value r.p.m. 0.988. Thus, the correction of the mean value of n_{TK} by $n_{TK}^{correct} = 0.994 - 0.988 = 0.006$.

According to [8, 9], the refined value of the specific fuel consumption C_e for the corrected value of the gas generator rotor r.p.m $n_{TK} = n_{TKopt} = 0.988$. The debugged engine (after adjusting the parameter n_{TK}) will correspond to the parameters $n_{TKopt} = 0.988$, $C_{e\,opt} = 0.971$. To demonstrate the use of a feed forward neural network with the use of adaptive elements for solving the problem of debugging the parameters of helicopters aircraft gas turbine engines in flight mode, a two-dimensional case of classification is considered. The practical application of this problem lies in the fact that one of two narrow-band random processes (RP) [27] is observed by means of a quadrature demodulator. It is known that the probability density of each of the processes is described by the expression [28, 29]:

$$p(N,C)=\frac{1}{\sqrt{2\pi}\cdot\sigma_N\cdot\sigma_C}\cdot e^{-\left(\frac{(N-m_N)^2}{2\sigma_N^2}+\frac{(C-m_C)^2}{2\sigma_C^2}\right)}; \qquad (39)$$

where σ_N, σ_C – variances; m_N, m_C – mathematical expectations of the components of RP; N– corresponds to the values of the gas generator rotor r.p.m n_{TK}; C – corresponds to the values of the specific fuel consumption C_e.

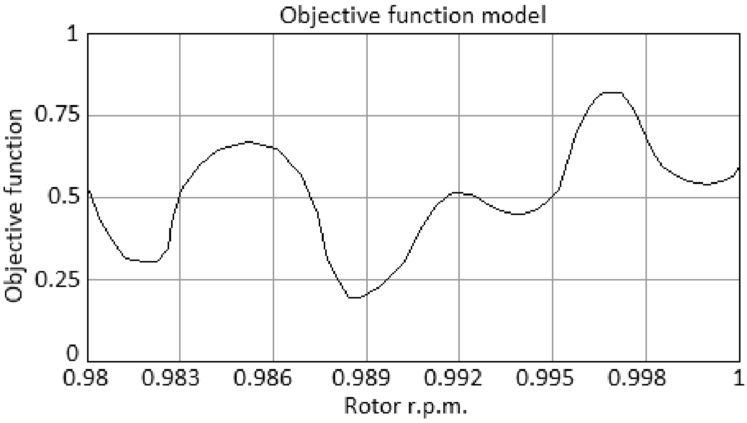
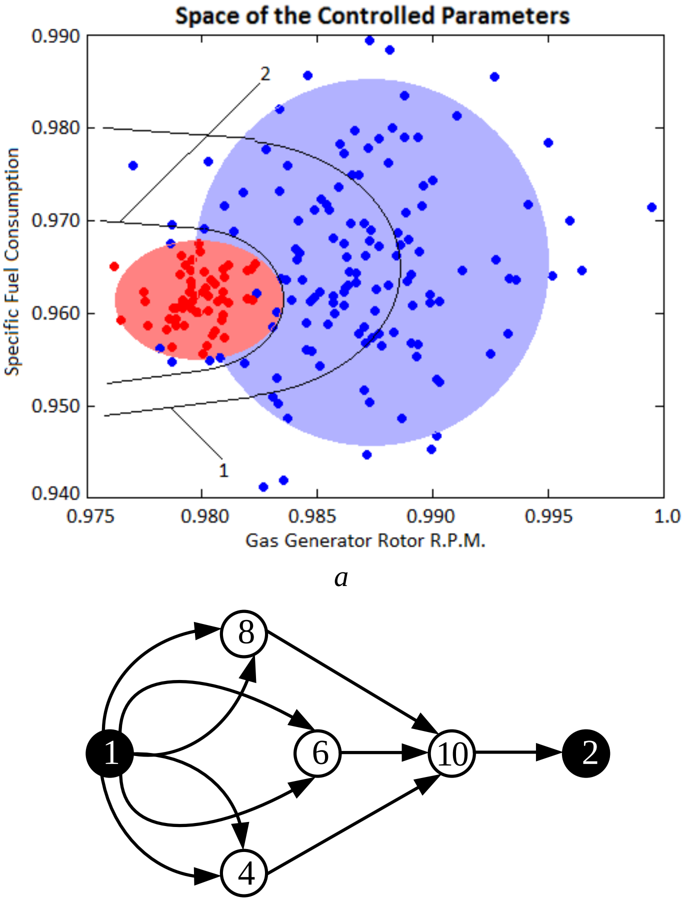


Figure 11: Graph of the dependence of the objective function from the value of gas generator rotor r.p.m.



a

b

Figure 12: Results: *a* – data of two classes (blue area – allowed values n_{TK} and C_e; red area – invalid values n_{TK} and C_e) and boundary lines at levels 0.983 (Limit value n_{TK}), 0.988 (Value n_{TKopt}); *b* – neural network for solving the classification problem

The input data for the neural network are the coordinates of points (fig. 12, *a*) on the plane $\{I, Q\}$ belonging to one of two distinguishable classes corresponding to two random processes, the parameters of expression (39) for which are different. The network output must determine whether the point belongs to the first or second class. A network was created with two input neurons, two neurons in a hidden layer with sigmoid activation functions. The output neuron of the network also has a sigmoid activation function, which is necessary to obtain a limited output signal. It is assumed that the output signal of the network will be close to zero if the point belongs to the class "A" and close to one if the point belongs to the class "B". The created neural network is shown in fig. 12, *b*.

9. Conclusions

As a result of the research, the scientific and practical problem of debugging (adjusting) of helicopters aircraft gas turbine engines parameters (using the example of the TV3-117 aircraft engine) in flight modes using neural network technologies has been solved.

The use of the neural network apparatus turns out to be effective in solving a wide range of poorly formalized problems associated with debugging of helicopters aircraft gas turbine engines parameters.

The results of solving the problem of debugging the parameters of aircraft gas turbine engines show that the process of debugging the parameters is easily formalized in a neural network basis, and to calculate the required value of r.p.m, an adjustment curve can be used, built on the basis of training the neural network according of aircraft gas turbine engines throttle characteristics.

The work uses a neural network of direct signal transmission, built on the basis of simple adaptive elements. A software prototype has been developed that implements adaptive elements within the framework of an object-oriented approach, as well as a specialized class library in Python for working in the sage environment. Thus, the obtained results of theoretical calculations and experimental research allowed to formulate the following position of scientific novelty of the work, which is as follows: further developed neural network method of adjusting the parameters of aircraft engines, which, through the use of universal mathematical model of adjusting the parameters of helicopters aircraft engines, the method of direct propagation of A.M. Lyapunov, as well as a neural network of direct propagation with adaptive elements, allowed to adjust the value of r.p.m. in accordance with the areas of permitted and prohibited values.

10. References

[1] K. Moon, A. A. Yakovlev, A comparative statistical analysis of global trends in civil helicopter accidents in the U.S., the EU, and the CIS, 18th International Conference "Aviation and Cosmonautics" (AviaSpace-2019) (2020) doi: https://doi.org/10.1088/1757-899X/868/1/012020 URL: https://iopscience.iop.org/article/10.1088/1757-899X/868/1/012020/pdf

[2] T.-O. Nævestad, R. J. Bye, S. Antonsen, S. H. Berge, I. S. Hesjevoll, B. Elvebakk, Examining the most accident-prone sector within commercial aviation: Why do accidents with light inland helicopters occur, and how can we improve safety?, Safety Science, vol. 139 (2021), 105235. doi: https://doi.org/10.1016/j.ssci.2021.105235

[3] S. P. Baker, J. G. Grabowski, R. S. Dodd, D. F. Shanahan, M. W. Lamb, G. H. Li, EMS helicopter crashes: What influences fatal outcome?, Annals of Emergency Medicine, vol. 47, issue 4 (2006), 351–356. doi: https://doi.org/10.1016/j.annemergmed.2005.11.018

[4] M. Guida, F. Marulo, S. Abrate, Advances in crash dynamics for aircraft safety, Progress in Aerospace Sciences, vol. 98 (2018) 106–123. doi: https://doi.org/10.1016/j.paerosci.2018.03.008

[5] H. H. Omar, V. S. Kuz'michev, A. Y. Tkachenko, Optimization the main thermodynamics parameters of the aviation turbofan engines with heat recovery in the aircraft system, Journal of Physics: Conference Series, vol. 1945 (2021) 012105. doi: https://doi.org/10.1088/1742-6596/1745/1/012105

[6] S. Chauhan, G. Jims, J. Wessley, Thermodynamic performance analysis of a micro turbojet engine for UAV and drone propulsion, International Journal of Recent Technology and Engineering (IJRTE), vol. 7, issue 5S3 (2019) URL: https://www.ijrte.org/wp-content/uploads/papers/v7i5s3/E11000275S19.pdf

[7] O. D. Lyantsev, A. V. Kazantsev, A. I. Abdulnagimov, Identification method for nonlinear dynamic models of gas turbine engines on acceleration mode, Procedia Engineering, vol. 176 (2017) 409–415. doi: https://doi.org/10.1016/j.proeng.2017.02.339

[8] S. V. Zhernakov, A. V. Kinarskij, Debugging the parameters of an aircraft gas turbine engine based on neural network technologies, Bulletin of USATU, vol. 17, no 5 (58) (2013) 26–30.

[9] Yu. Shmelov, S. Vladov, O. Kryshan, S. Gvozdik, Application of neural networks in the problem of setting of the parameters of the aviation engine TV3-117 in flight modes, Visnyk of Kherson National Technical University, no. 4 (67) (2018) 126–132.

[10] F. Li, W. X. Zheng, S. Xu., D. Yuan, A novel ε-dependent Lyapunov function and its application to singularly perturbed systems, Automatica, vol. 133 (2021) 109749. doi: https://doi.org/10.1016/j.automatica.2021.109749

[11] S. V. Rakovic, Robust Control Minkowski-Lyapunov functions, Automatica, vol. 125 (2021) 109437. doi: https://doi.org/10.1016/j.automatica.2020.109437

[12] M. Jin, M. Brake, H. Song Comparison of nonlinear system identification methods for free decay measurements with application to jointed structures, Journal of Sound and Vibration, vol. 453 (2019) 268–293. doi: https://doi.org/10.1016/j.jsv.2019.04.021

[13] V. Yu. Tetter, E. V. Shendaleva, Identification of the characteristics of converters when adjusting the fuel regulators of the ACS GTE after assembly, Assembly in mechanical engineering, instrument making, no. 4 (2004) 7–14.

[14] E. N. Efimov, T. Ya. Shevgunov, Development of feedforward neural networks using adaptive elements, Journal of Radio Electronics, no. 8 (2012). URL: http://jre.cplire.ru/win/aug12/4/text.html

[15] X. Wang, X. Lin, X. Dang, Supervised learning in spiking neural networks: A review of algorithms and evaluations, Neural Networks, vol. 125 (2020) 258–280. doi: https://doi.org/10.1016/j.neunet.2020.02.011

[16] Aggarwal C. C. (2018) Neural Networks and Deep Learning, Cham, Springer, 497 p.

[17] Wlodarczak P. (2019) Machine Learning and its Applications. Florida, CRC Press, 205 p.

[18] Vladov S., Shmelov Yu. and Shmelova T. (2020) Control and diagnostics of TV3-117 aircraft engine technical state in flight modes using neural network technologies, Kremenchuk, Novabook, 200 p.

[19] J. An, F. Liu, J. Zhao, F. Shen, R. Li., K. Gao, IC neuron: An efficient unit to construct neural networks, Neural Networks, vol. 145 (2020) 177–188. DOI: 10.1016/j.neunet.2021.10.005

[20] S. Chakraverty, D. M. Sahoo, N. R. Mahato, McCulloch–Pitts Neural Network Model, Concepts of Soft Computing (2019) 167–173. doi: https://doi.org/10.1007/978-981-13-7430-2_11

[21] A. Abdalazeim, F. Meziane, A review of the generation of requirements specification in natural language using objects UML models and domain ontology, Procedia Computer Science, vol. 189 (2021) 328–334. DOI: 10.1016/j.procs.2021.05.102

[22] A. Zollanvari, E. R. Dougherty, Moments and root-mean-square error of the Bayesian MMSE estimator of classification error in the Gaussian model, Pattern Recognition, vol. 47, issue 6 (2014) 2178–2192. doi: https://doi.org/10.1016/j.patcog.2013.11.022

[23] P. Shukl, B. Singh, Delta-Bar-Delta neural-network-based control approach for power quality improvement of Solar-PV-Interfaced distribution system, IEEE Transactions on Industrial Informatics, vol. 16, issue 2 (2020), 790–801. doi: https://doi.org/10.1109/TII.2019.2923567

[24] M. A. Konovalyuk, A. A. Gorbunova, Yu. V. Kuznecov, A. B. Baev, Algorithm for extracting information from a complex radar image of a complex target, Radar and radio communication: materials of the 4th all-Russian conference (2010) 25–29.

[25] S. Vladov, Yu. Shmelov, K. Kotliarov, S. Hrybanova, O. Husarova, I. Derevyanko, L. Chyzhova, Onboard parameter identification method of the TV3-117 aircraft engine of the neural network technologies, Transactions of Kremenchuk Mykhailo Ostrohradskyi National University, issue 5/2019 (118) (2019) 90–96. doi: https://doi.org/10.30929/1995-0519.2019.5.90-96

[26] S. V. Zhernakov, Algorithms for control and diagnostics of an aviation gas turbine engine under conditions of onboard implementation based on neural network technology", Bulletin of USATU, vol. 14, no. 3 (38) (2010) 42–56.

[27] G. Picci, B. Zhu, Bayesian Frequency Estimation on Narrow Bands, IFAC-PapersOnLine, vol. 54, issue 7 (2021), 108–113. doi: https://doi.org/10.1016/j.ifacol.2021.08.343

[28] A. Vexler, Univariate likelihood projections and characterizations of the multivariate normal distribution, Journal of Multivariate Analysis, vol. 179 (2020). 104643. DOI: 10.1016/j.jmva.2020.104643

[29] A. Menezes, J. Mazucheli, J. Cardoso, S. Chakraborty, The transmuted half-normal distribution with application to precipitation data, Pesquisa Operacional, vol. 40 (1) (2020). DOI: 10.1590/0101-7438.2020.040.00216792. URL: https://www.scielo.br/j/pope/a/CWztkckFRcGwLTtkYHmnXCn/?format=pdf&lang=en