

Article

Onboard Neuro-Fuzzy Adaptive Helicopter Turboshift Engine Automatic Control System

Serhii Vladov ¹, Maryna Bulakh ^{2,*}, Victoria Vysotska ^{3,4} and Ruslan Yakovliev ¹

¹ Kremenchuk Flight College of Kharkiv National University of Internal Affairs, 17/6 Peremohy Street, 39605 Kremenchuk, Ukraine

² Department of Industrial Processes Computerization and Robotization, Faculty of Mechanics and Technology, Rzeszow University of Technology, 4 Kwiatkowskiego Street, 37-450 Stalowa Wola, Poland

³ Information Systems and Networks Department, Lviv Polytechnic National University, 12 Bandera Street, 79013 Lviv, Ukraine

⁴ Institute of Computer Science, Osnabrück University, 1 Friedrich-Janssen-Street, 49076 Osnabrück, Germany

* Correspondence: m.bulakh@prz.edu.pl

Abstract: A modified onboard neuro-fuzzy adaptive (NFA) helicopter turboshift engine (HTE) automatic control system (ACS) is proposed, which is based on a circuit consisting of a research object, a regulator, an emulator, a compensator, and an observer unit. In this scheme, it is proposed to use the proposed AFNN six-layer hybrid neuro-fuzzy network (NFN) with Sugeno fuzzy inference and a Gaussian membership function for fuzzy variables, which makes it possible to reduce the HTE fuel consumption parameter transient process regulation time by 15.0 times compared with the use of a traditional system automatic control (clear control), 17.5 times compared with the use of a fuzzy ACS (fuzzy control), and 11.25 times compared with the use of a neuro-fuzzy reconfigured ACS based on an ANFIS five-layer hybrid NFN. By applying the Lyapunov method as a criterion, its system stability is proven at any time, with the exception of the initial time, since at the initial time the system is in an equilibrium state. The use of the six-layer ANFF NFN made it possible to reduce the I and II types of error in the HTE fuel consumption controlling task by 1.36...2.06 times compared with the five-layer ANFIS NFN. This work also proposes an AFNN six-layer hybrid NFN training algorithm, which, due to adaptive elements, allows one to change its parameters and settings in real time based on changing conditions or external influences and, as a result, achieve an accuracy of up to 99.98% in the HTE fuel consumption controlling task and reduce losses to 0.2%.

Keywords: helicopters turboshift engines; automatic control system; neuro-fuzzy network; bell-shaped membership function; ANFF; ANFIS; fuel consumption; transient process



Citation: Vladov, S.; Bulakh, M.; Vysotska, V.; Yakovliev, R. Onboard Neuro-Fuzzy Adaptive Helicopter Turboshift Engine Automatic Control System. *Energies* **2024**, *17*, 4195. <https://doi.org/10.3390/en17164195>

Academic Editor: Constantine D. Rakopoulos

Received: 28 May 2024

Revised: 17 August 2024

Accepted: 19 August 2024

Published: 22 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Relevance of the Research

Most dynamic systems operate under uncertainty conditions, where there are complex and poorly understood relations between various technological factors, together with the presence of external influences and random interference, as well as nonlinear elements that make the conventional linear adaptive control algorithms difficult to apply [1,2]. In such conditions, the development of effective control strategies becomes a particularly important task. One approach to achieving this task is the hybrid use of neural networks (NNs) [3,4] and fuzzy logic (FL) [5,6]. NNs can train from existing data and identify complex nonlinear relations between the in- and output variables of a system. On the other hand, FL allows us to describe fuzzy or uncertain quantities and relations between them, which makes control models more flexible and adaptive to changing conditions. With the NN and FL hybrid application, the control object and its regulator are described by fuzzy adaptive models. These models can change their structure and parameters during system operation depending on current conditions and control requirements. This approach

makes it possible to effectively manage the advanced dynamic systems without requiring precise knowledge of their internal structure or the nature of the influencing external factors [7,8]. The use of hybrid control systems with NNs and FL can lead to significant improvements in the performance and reliability of controlled objects under conditions of environmental uncertainty and variability. Such systems can be successfully used in various fields, including aviation [9,10], industry [11,12], transport [13,14], medicine [15,16], and many others [17–20].

The relevance of this work lies in addressing the challenges faced by most dynamic systems operating under uncertainty, where complex and poorly understood relationships exist between various technological factors, external influences, and random interference, making conventional linear adaptive control algorithms difficult to apply; the proposed hybrid approach, combining neural networks and fuzzy logic, offers a flexible and adaptive solution for effective management in dynamic environments across various fields.

1.2. State of the Art

Currently, hybrid control integrating NNs and FL is of particular relevance in the gas turbine engine (GTE) controlling context, including HTE. GTEs are complex systems operating under conditions of high dynamics and uncertainty [20,21]. This is where the need arises to develop and apply effective control methods that can adapt to different operating scenarios and ensure high engine performance and reliability. Hybrid control systems, blending the benefits of NNs and FL, provide ample opportunities for fulfilling such tasks [22,23].

For example, in [24], researches employed a fault diagnosis and control system for operations using hybrid multi-mode approaches of the machine learning for monitoring the system health status, using the integration of recurrent feature generation based on NNs and diagnostic modules based on self-organizing maps, including the development of systematic clustering and modeling methodology to create reliable diagnostic modules and check the system efficiency using data from sensors as an example, in relation to the compressor and GTE. A key disadvantage is the limitation of multi-mode and parallel faults, which can pose serious obstacles to the development of reliable diagnostic techniques.

In [25], the authors proposed a model-based learning control method to improve the control quality during the GTE afterburning phase, including offline training and training modules, a power arm angle reference path module, and a built-in nonlinear online inference module, which effectively reduces the total pressure fluctuation on turbine outlet, increases the fan surge margin, and improves thrust linearity. The key disadvantage of this method is its dependence on the accuracy of the model and the possibility of errors due to imperfections in the system model representation.

In [26,27], a modified closed onboard neural network HTE ACS was developed, supplemented with plug-in adaptive control software modules, including a signal adaptation module, a parametric adaptation module, a linear model submodule, and a custom submodule model, differing from current systems, by separating into distinct links for GTE and fuel metering unit (FMU); however, it overlooks the transient processes' synchronism in the fuel system and the engine itself, which can lead to problems with system overshoot. Overshoot of the developed HTE ACS can cause sudden changes to occur under conditions in control inputs or flight conditions, such as changes in speed, altitude, angle of attack, or during sudden disturbances, such as turbulence of air flows or sudden changes in aerodynamic conditions.

In order to possibly eliminate the overshooting problem of the HTE ACS, an alternative method for its construction was developed in [28], which makes it possible to ensure specified stability indicators by HTE reconfiguring ACS using hybrid NFNs of the ANFIS type with a zero-order Takashi–Sugeno–Kanga training algorithm. This method is adapted to control HTE at flight condition with a minimum control error not exceeding 0.4%. The work also for the first time proposed the use of bell-shaped membership functions of linguistic variables to describe the HTE thermogas-dynamic parameters (TDP), as well as

the linguistic expression “about” in a fuzzy knowledge base, which makes it possible to adjust parameter values under uncertainties with an accuracy of up to 99.6% (the maximum control error does not exceed 0.4%) in conditions of changing factors, such as errors, flight conditions, and helicopter operational status.

A key disadvantage [28] that requires improvement is the need for further research and control method development that would ensure system stability and minimal control error in a wide range of operating conditions, including various flight modes and variability in the helicopter operational status.

To eliminate this key drawback, and thus to ensure system stability and minimal control error in various helicopter operating conditions, it is advisable to use an NFA ACS [29], in which disturbing influences are compensated, and the control error is controlled. A neuro-fuzzy model based on the Sugeno model has the potential to adapt to changing conditions and compensate for the disturbance effects. In this case, an emulator representing an HTE model is used to configure the controller, which further increases the system efficiency due to more accurate adjustment of control parameters based on real data. However, to further improve the system, it is necessary to conduct additional research and develop methods that allow more effective HTE supervision under diverse operating conditions and with disturbances various types.

1.3. Main Attributes of the Research

The object of the research is the HTE controls systems.

The subject of the research includes the NFA HTE ACS.

The research aims to develop a NFA HTE ACS to ensure stability and minimal control error under diverse operating conditions based on an adaptive identifier for a neuro-fuzzy ACS under uncertainty conditions.

To achieve this aim, the following tasks are performed:

1. Development of an HTE control algorithm, which provides an acceptable control error when fulfilling the restrictions.
2. Development of structural and parametric identification algorithms in real time, combining the linear equation identifying coefficients method and interactive adaptation theory technique.
3. Development of a hybrid model based on NNs and FL to improve the efficiency of HTE control solving the task under conditions of uncertainty.
4. Modification of the Sugeno fuzzy model to improve the complex dynamic objects control efficiency by introducing algorithms for structural and parametric identification in real time.
5. Improving the training process of a fuzzy network by applying a modified interactive adaptation method.
6. Conducting a computational experiment consisting of HTE fuel consumption parameter transient process modeling during its stepwise change.
7. Determination of ACS quality indicators (stability, control accuracy, control speed, resistance to disturbances, minimal overshoot, and time characteristics of transient processes) in one of HTE's operating modes (for example, in the nominal mode).
8. Conducting a comparison of the quality indicators obtained for the developed HTE ACS to demonstrate superiority in control efficiency, stability, and control accuracy in comparison with alternative methods.

The research results make a significant contribution to the development of neuro-fuzzy control systems for complex dynamic objects, such as helicopter turboshaft engines (HTEs), that can operate under changing factors. These findings will interest not only specialists in neuro-fuzzy systems but also developers of HTE control systems and experts involved in controlling complex dynamic objects under uncertainty. In summary, the main contribution of this research is the advancement of adaptive control techniques that enhance system reliability and performance in dynamic environments.

1.4. Structure of the Article

The article consists of an introduction, the main part (sections “Materials and Methods”, “Results”, “Discussion”), conclusions, and references. The section “Materials and Methods” consists of the following paragraphs: “2.1. Research object model”, “2.2. Compensator model”, “2.3. Emulator model”, “2.4. Controller model”, “2.5. Clarifying the emulator model taking into account the controller model”, “2.6. Six-layer NFN AFNN training algorithm”, “2.7. Observer block introduction into an NFA modified closed onboard TE ACS”. In section “2.1. Research object model”, a mathematical model and a structural diagram of a neuro-fuzzy adaptive modified closed system of automatic control of complex dynamic objects are developed. Sections 2.2–2.6 are devoted to the development of the complex dynamic objects neuro-fuzzy adaptive modified closed automatic control system component parts mathematical models. Section “2.6. Six-layer NFN AFNN training algorithm” describes the algorithm for training the neuro-fuzzy network, which is used to implement the developed system. Section “2.7. Observer block introduction into an NFA modified closed onboard TE ACS” develops the observer block mathematical model. The “Results” section describes the results of the HTE thermogas-dynamic parameters modeling transient processes. The “Discussion” section calculates the obtained results quality metrics and describes a comparison with the analogues.

2. Materials and Methods

2.1. Research Object Model

The HTE ACS main task is to ensure optimal regulation of engine operation, including fuel supply regulation depending on the gas-generator rotor r.p.m. n_{TC} achieved at various stages of engine operation, including idle mode [30,31], provided that the control object (HTE) dynamics is presented in the modified nonlinear difference equation form:

$$y(i+1) = f(y(i), \dots, y(i-r), x(i), \dots, x(i-s), u(i), \dots, u(i-q), d(i)), \quad (1)$$

in which the variable $d(i)$ is an additional parameter that takes into account dynamic changes in external conditions or other factors that may influence the ACS; $i = 0, 1, 2, \dots, N$ is the current discrete time; $y(i)$ is the output signal; $x(i) = (x_1(i) \dots x_k(i))$ is the disturbing influences vector; $u(i)$ is the control; and $f(\bullet)$ is some nonlinear function having known orders r, s, q . The disturbance vector $x(i)$ and control $u(i)$ are limited at any time, that is:

$$\begin{cases} u^{min} \leq u_i \leq u^{max}, \\ x^{min} \leq x_i \leq x^{max}, \\ d^{min} \leq d_i \leq d^{max}. \end{cases} \quad (2)$$

The solution to the nonlinear difference Equation (1) is the HTE control algorithm $y(i+1)$, which provides an acceptable control error $e(i+1) = y_{nom} - y(i+1)$ (y_{nom} is the nominal output value), taking into account the control $u(i)$ and the object $y(i)$ current state, as well as disturbances $x(i)$ and $d(i)$, using the following expression:

$$u(i) = f(y(i), x(i), d(i)). \quad (3)$$

To solve the HTE controlling task, in [28] a hybrid NFN of the ANFIS type (Figure 1) use was proposed, which contains 5 layers: the first layer is the nonlinear object inputs under research (HTE); the second layer is the layer of fuzzy terms, which are used in the HTE fuzzy knowledge base; the third layer is the fuzzy knowledge base (fuzzy rules) conjunction lines; the fourth layer is the output variable d_j classes, which is associated with a reference sample known in the fuzzy knowledge base; and the fifth layer is the defuzzification layer, that is, converting the fuzzy output into a crisp number.

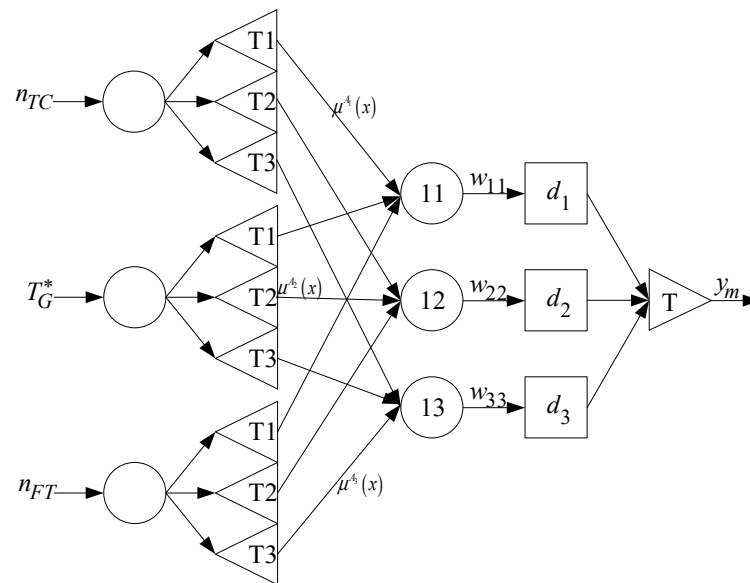


Figure 1. Reconfigured modified closed TE ACS model.

An key disadvantage of the ANFIS is its limited ability to adapt to complex and rapidly changing environmental conditions, which can lead to insufficient control accuracy in some scenarios. To eliminate these shortcomings, it is advisable to use an NFA ACS (Figure 2), proposed in [29], in which the influence of disturbing influences is largely eliminated by a compensator, and the control error $e(i + 1)$ is eliminated by a controller, which is an adjusted emulator of an object model under research (HTE). This allows for more precise control in changing environmental conditions and improves the control system efficiency, especially in the context of dynamic objects such as HTE.

In Figure 2, the controller manages input signals for the system based on predefined rules and membership functions, ensuring the desired system behavior in changing conditions. The emulator simulates the dynamics of the controlled object, allowing the system to predict real object behavior and adjust controller actions for optimal performance. The compensator adjusts deviations from the desired system state, using feedback to minimize errors and stabilize system operation. The observer block evaluates the system state in real time, using sensor data and mathematical models to provide accurate information to the controller and other system components.

A neuro-fuzzy control system integrates principles from neural networks and fuzzy logic for effective management of complex dynamic objects. The controller makes decisions based on fuzzy rules that adapt to changes in the external environment. The emulator models object behavior, predicting responses to various influences, allowing the controller to make more accurate decisions. The compensator corrects deviations from the desired state, adjusting system actions to reduce errors. The observer block monitors object state in real time, providing up-to-date data for all system elements, enhancing their interaction and increasing control accuracy. Together, these components provide reliable and adaptive management capable of handling uncertainty and nonlinearity in controlled processes.

To account for the real-time aspect in a neuro-fuzzy control system, it is essential to implement efficient algorithms that process inputs and generate control signals within strict time constraints, ensuring timely responses to dynamic changes in the environment. This can be achieved by optimizing computational resources and employing parallel processing techniques to handle the system's complex calculations without compromising speed or accuracy [32,33].

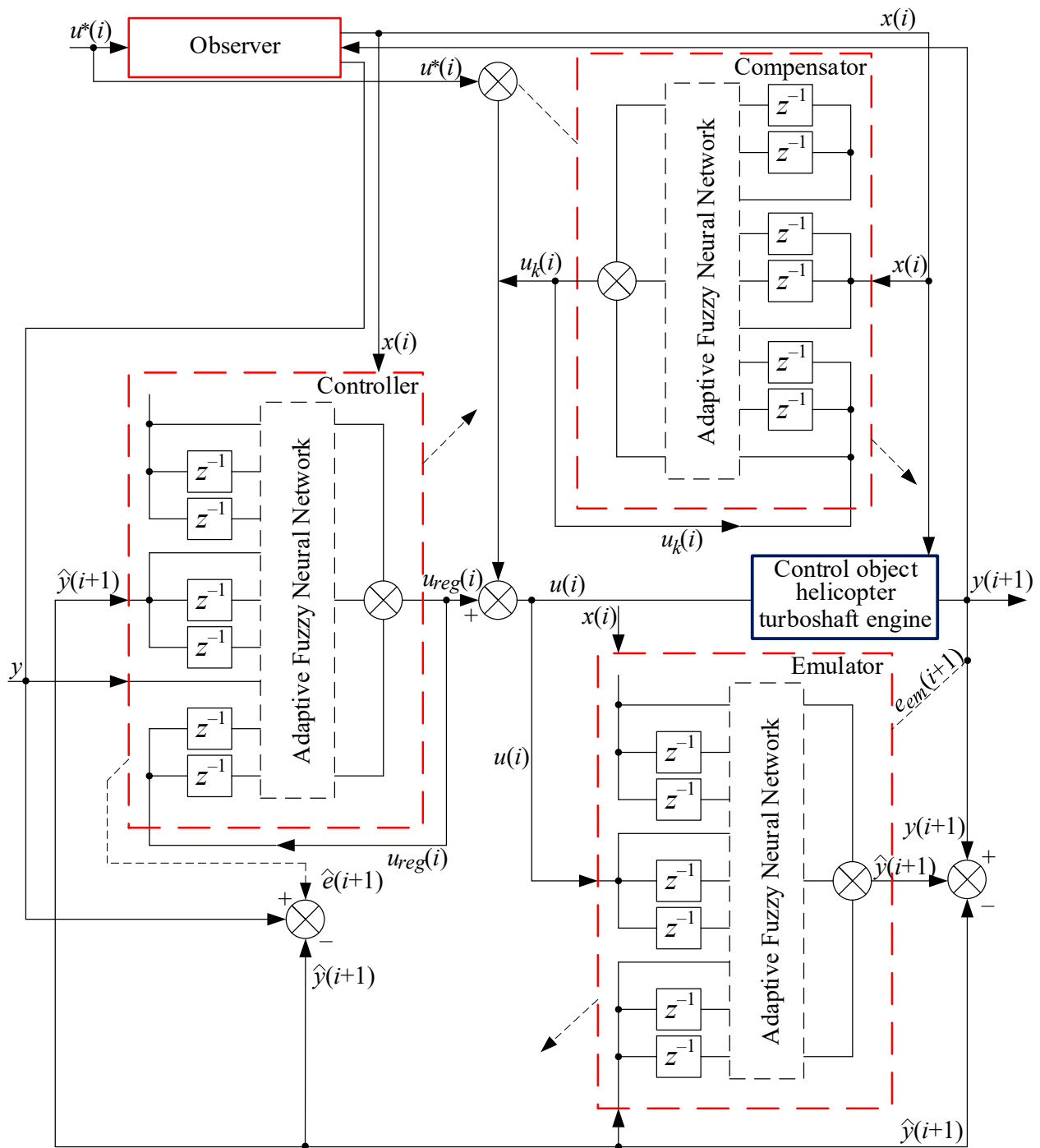


Figure 2. Structure of an NFA modified closed TE ACS.

The controller, compensator, and emulator contain delay elements $z(\tau) = x(t - \tau)$, $\tau = 1, 2, \dots$, which form the input and output variables with a delay. Similar to (1), the compensator is described by a nonlinear difference equation form:

$$u_k(i) = f_k(u_k(i-1), \dots, u_k(i-q_k), x(i), x(i-1), \dots, x(i-s_k), c_k) + \epsilon_k(i), \quad (4)$$

having orders q_k and s_k , differing in the general case from q and s , while $u_k(i-1) \dots u_k(i-q_k)$ are scalar control actions, $x(i)$ is a dimension v input disturbances vector, and c_k is a parameters settings vector. The orders q_k and s_k indicate feedback delay and input signal delay, respectively. The introduction of a noise term (error) $\epsilon_k(i)$ into the compensator Equation (4)

is justified by the need to take into account non-idealities and variable environmental conditions. In real-world conditions, measurements may be subject to noise, interference, or other external factors that may cause the compensator to operate inaccurately. In addition, even when using an accurate mathematical model, modeling errors and internal inaccuracies in compensator components, such as imperfections in design or calibration, are inevitable. The noise term introduction allows us to take these factors into account and make the compensator model more realistic, ensuring its better adaptability to real operating conditions.

According to [29], a replacement is introduced in (4):

$$u_k(i - 1), \dots, u_k(i - q_k), x(i), x(i - 1), \dots, x(i - s_k) = x_k(i) = \{x_{k1}(i) \dots x_{km}(i)\}, \quad (5)$$

where $m = q_k + v \cdot (s_k + 1)$.

Then (4) takes the form:

$$u_k(i) = f_k(x_k(i), c_k) + \epsilon_k(i). \quad (6)$$

Based on [29], the Sugeno fuzzy model, represented by rules set, is used as the difference Equation (6) to describe the compensator:

$$\begin{aligned} &\text{if } x_{k1}(i), \text{ there is } X_{k1}^\theta \\ &x_{k2}(i) \text{ there is } X_{k2}^\theta \dots x_{km}(i) \text{ there is } X_{km}^\theta \\ \text{then } u_k^\theta(i) &= b_{k0}^\theta + b_{k1}^\theta \cdot x_{k1}(i) + \dots + b_{km}^\theta \cdot x_{km}(i) + \epsilon_k(i) \end{aligned} \quad (7)$$

with fuzzy sets $X_{kl}^\theta, l = \overline{1 \dots m}, \theta = \overline{1 \dots n}$, and a linear dependence connecting inputs $x_k(i) = (x_{k1}(i), x_{k1}(i), \dots, x_{km}(i))$, output $u_k(i)$, and noise term (error) $\epsilon_k(i)$.

The main characteristic that defines the fuzzy set X_k is its membership function $\mu(X_k(x_k))$, which, according to [28], is a bell-shaped membership function (Figure 3) of the form:

$$\mu(X_k(x_k)) = \frac{1}{1 + \left(\frac{x_k - d_{k1}}{d_{k2}}\right)^2}, \quad (8)$$

where d_{k1} and d_{k2} are the bell-shaped membership function parameters.

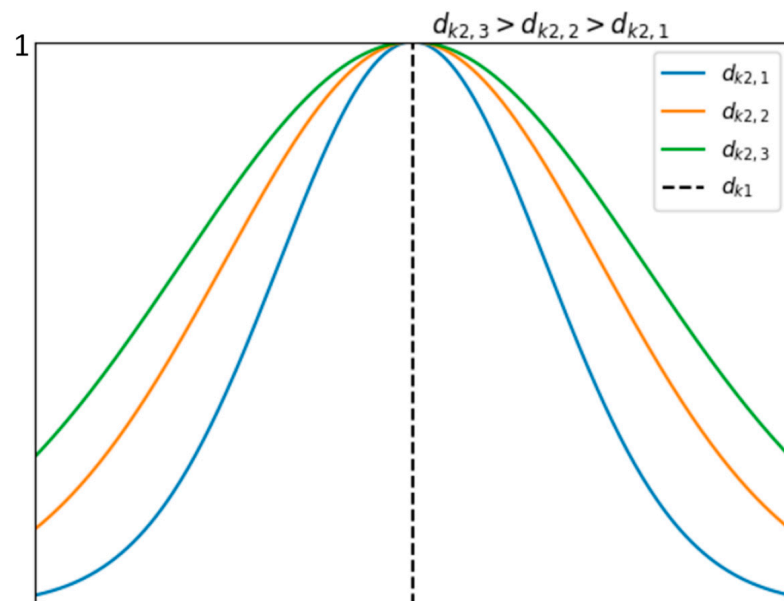


Figure 3. Bell-shaped membership function general view.

According to [28], the fuzzy variable bell-shaped membership function use is justified in cases where it is necessary to more flexibly control the membership shape and distribu-

tion around the central value. Unlike the sigmoid function proposed in [29], which has a more limited range and faster saturation, the bell function allows us to more precisely tune the curve shape and adapt it to the system-specific requirements. This is especially useful in cases where there is a need to account for data uncertainty and varying degrees of fuzziness, or when there are multiple peaks in the distribution.

The choice of the bell-shaped membership function, as opposed to other types such as sigmoid functions, is primarily due to its superior flexibility in modeling fuzzy variables. The bell-shaped function, characterized by its smooth, symmetric curve, offers enhanced control over the shape and distribution around the central value, making it particularly effective in scenarios where precise tuning is required [34]. This function's broader range and gradual saturation provide better adaptation to varying degrees of data uncertainty and fuzziness compared to sigmoid functions, which have a more constrained range and quicker saturation [35]. Furthermore, the bell-shaped membership function's ability to accommodate multiple peaks in the distribution is crucial when dealing with complex systems exhibiting diverse operational conditions and varying degrees of uncertainty [36]. This flexibility ensures a more accurate representation of the system's behavior and improves the overall performance of the neuro-fuzzy control model, aligning it more closely with specific system requirements and contributing to more reliable and adaptable control strategies.

When determining the output $u_k(i)$ mechanism according to the fuzzy model (8) when specifying inputs $X_{kl}^0(i)$ at time $i = 1, 2, \dots, N$, membership functions $\mu(X_{kl}^0(x_{kl}(i)))$, coefficients $b_{k0}^\theta, b_{k1}^\theta, \dots, b_{km}^\theta$, and equation $u_k^\theta(i) = b_{k0}^\theta + b_{k1}^\theta \cdot x_{k1}(i) + \dots + b_{km}^\theta \cdot x_{km}(i) + \epsilon_k(i)$, taking into account [28,29], it is advisable to present it in an AFNN-type (adaptive fuzzy neural network) six-layer NFN form (Figure 4) [37].

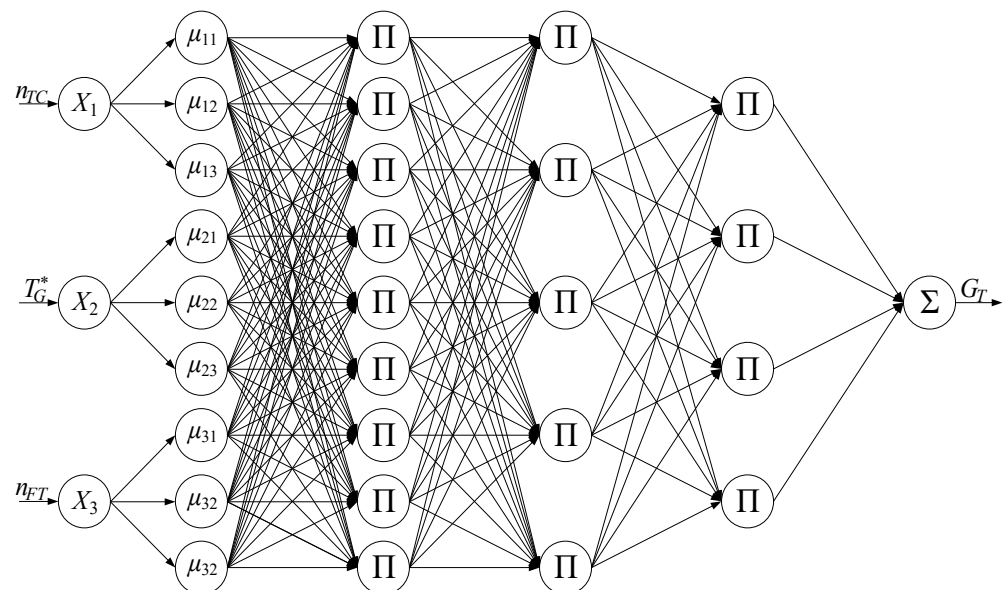


Figure 4. Structure of the proposed AFNN-type six-layer hybrid NFN.

AFNN, based on the Sugeno model, is more appropriate for the neuro-fuzzy HTE ACS constructing task than ANFIS [28] or TSK models [38], due to its ability to handle uncertainty and nonlinearity in the system using a combination of FL and NNs. The Sugeno model provides a more flexible mechanism for describing fuzzy rules, which allows the system to better adapt to different HTE operating conditions. In addition, an AFNN using the Sugeno model can better adapt to changing conditions and requirements, thanks to a more complex network structure and the ability to train not only membership function parameters, but also neuron weights, which reduces the need to manually tune the system for each specific situation.

The Sugeno-based AFNN model offers significant advantages over traditional models like ANFIS or TSK for HTE neuro-fuzzy control systems. Its ability to handle uncertainty and nonlinearity through a combination of FL and NNs provides a more flexible mechanism for describing fuzzy rules, enabling better adaptation to varying operational conditions. Additionally, the Sugeno-based AFNN can more effectively adjust to changing conditions and requirements due to its complex network structure and the capability to train both membership function parameters and neuron weights, reducing the need for manual system adjustments [39,40].

As mentioned above, AFNN based on the Sugeno model consists of six layers. The first layer (Input Layer) accepts model input data, which are the HTE TDP numerical values, recorded on board the helicopter at flight mode: gas-generator rotor r.p.m. n_{TC} , free turbine rotor speed n_{FT} , and gas temperature in front of the compressor turbine T_G^* .

In the second layer (Fuzzy Layer), the input data are converted into fuzzy values using membership functions (8) by calculating the membership degrees $X_{k1}^\theta X_{k1}^\theta(x_{k1}^\theta(i)) \dots X_{km}^\theta(x_{km}^\theta(i))$ for θ -th rules.

In the third layer (First neural layer), the truth values of premises w_k^θ are calculated by algebraic multiplication:

$$w_k^\theta = X_{k1}^\theta(x_{k1}^\theta(i)) \cdot X_{k2}^\theta(x_{k2}^\theta(i)) \cdot \dots \cdot X_{km}^\theta(x_{km}^\theta(i)). \quad (9)$$

The premises w_k^θ truth values calculation is a critical stage that determines the contribution of each premises to the corresponding rule activation. These values are each input variable X_k^θ membership degree's product for a given rule θ , which reflects the model confidence degree in fulfilling the input data rule given conditions. Thus, high values of w_k^θ indicate a stronger coincidence of the input variables with the rules, which makes it possible to effectively take into account fuzziness and uncertainty in the data and generate correct output recommendations or HTE ACS control signals.

In the fourth–second neural layer, relative normalized values are determined as:

$$\beta_k^\theta(i) = \alpha \cdot \frac{w_k^\theta(i)}{w_k^1(i) + w_k^2(i) + \dots + w_k^n(i)} = \alpha \cdot \frac{w_k^\theta(i)}{\sum_{j=1}^n w_k^j(i)}, \quad (10)$$

where α is a coefficient that can be adjusted during the network training process or determined based on additional expert information. The α coefficient allows more flexibility in each parcel contribution controlling to the normalized values, which can be useful if some parcels have greater meaning or significance than others.

In the fifth layer (Combining Layer), the outputs from the neural layer are combined using fuzzy methods or aggregation to produce the final model output. The $\beta_k^\theta(i)$ values are multiplied by the output values $u_k^0(i)$, calculated according to the equation $u_k^0(i) = b_{k0}^\theta + b_{k1}^\theta \cdot x_{k1}(i) + \dots + b_{km}^\theta \cdot x_{km}(i) + \epsilon_k(i)$ when setting the values $x_{k1}^0(i)$, $x_{k2}^0(i)$, \dots , $x_{km}^0(i)$.

$$\beta_k^\theta(i) \cdot u_k^0(i) = \alpha \cdot \frac{w_k^\theta(i)}{\sum_{j=1}^n w_k^j(i)} \cdot \left(\beta_{k0}^\theta + \sum_{j=1}^m \beta_{kj}^\theta \cdot x_{kj}^0(i) + \epsilon_k(i) \right), \quad (11)$$

where $w_k^\theta(i)$ is the k -th rule membership degree for the i -th input data, β_{k0}^θ is the θ -th rule bias parameter, β_{kj}^θ is the weighting coefficient for the j -th input in the θ -th rule, $x_{kj}^0(i)$ is the j -th input neuron for the k -th rule, and $\epsilon_k(i)$ is the additional error (noise).

Expression (11) is the product of the weights $\beta_k^\theta(i)$ and outputs $u_k^0(i)$ for each θ -th rule and each k -th input, which allows us to aggregate the contribution of each rule and its weighted output to the NFN final output.

The noise term (error) $\epsilon_k(i)$ is a random variable that represents the difference between the actual output value and its predicted or expected value at the i -th time moment.

Mathematically, it is described as a random variable with zero mean and some variance, which characterizes the spread between the actual and predicted values.

In the sixth layer (Output Layer), after multiplying and combining the outputs for all rules, the final NFN output signal is calculated, which is an aggregated representation of output values, reflecting the calculated predictive or control recommendations based on the input data and parameters of the trained network:

$$y = \sum_{\theta=1}^n \sum_{k=1}^m \beta_k^\theta(i) \cdot u_k^0(i). \tag{12}$$

2.2. Compensator Model

Based on [29], in the compensator fuzzy formulation, the vector tuning parameters c_k are formed from the coefficients b_{kl}^θ involved in the equations $u_k^\theta(i) = b_{k0}^\theta + b_{k1}^\theta \cdot x_{k1}(i) + \dots + b_{km}^\theta \cdot x_{km}(i) + \epsilon_k(i)$, as well as from the parameters $d_{k1,l}^\theta, d_{k2,l}^\theta$, related to the bell-shaped membership function shape. At the initial stage of setting up the compensator, the available data are used to determine the fuzzy rules number n using the structural identification algorithm ψ_n , the equations coefficients b_{kl}^θ using the parametric identification algorithm ψ_b , as well as the parameters $d_{k1,l}^\theta, d_{k2,l}^\theta$ using the backpropagation algorithm. In this case, data $x_k^*(i) = x_{k1}^*(i) \dots x_{km}^*(i), u^*(i)$ are used, for which the next output variable $y(i + 1)$ value is close to the nominal y_{nom} , such that it satisfies the condition [29,41]:

$$J^* = \frac{1}{N} \cdot \sum_{i=1}^N \left| \frac{y_{nom} - y(x_k^*(i), u^*)}{y_{nom}} \right| \leq J_{nom}. \tag{13}$$

where J_{nom} is the relative control error nominal value.

Thus, for the compensator model mathematical description, it is advisable to research the structural identification algorithms [37], parametric identification [38], and the back-propagation algorithm [39].

The algorithm for identifying the equation coefficients b_{kl}^θ , performed on the data set $x_k^*(i) = x_{k1}^*(i) \dots x_{km}^*(i), u^*(i)$, based on (7), (9), (10), (12) is represented in the forms:

$$u_k(i) = b_{k0}^1 \cdot \beta_k^1(i) + \dots + b_{k0}^n \cdot \beta_k^n(i) + b_{k1}^1 \cdot \beta_k^1(i) \cdot x_{k1}^*(i) + \dots + b_{k1}^n \cdot \beta_k^n(i) \cdot x_{k1}^*(i) + b_{km}^1 \cdot \beta_k^1(i) \cdot x_{km}^*(i) + \dots + b_{km}^n \cdot \beta_k^n(i) \cdot x_{km}^*(i) \tag{14}$$

or in vector form

$$u_k(i) = \overline{b_k^T} \cdot \widetilde{x}_k(i), \tag{15}$$

where $\widetilde{x}_k(i) = \beta_k^1(i) \dots \beta_k^n(i), \beta_k^1(i) \cdot x_{k1}^*(i), \dots, \beta_k^1(i) \cdot x_{km}^*(i) \dots \beta_k^i(i) \cdot x_{km}^*(i)^T$ is the extended input vector, and $\overline{b_k} = (b_{k0}^1 \dots b_{k0}^n, b_{k1}^1 \dots b_{k1}^n, \dots, b_{km}^1 \dots b_{km}^n)^T$ is the vector of adjustable coefficients. To determine the initial values of the elements of the vector $\overline{b_k}(0) = 0$ and the correction matrix of size $(m + 1) \times n \cdot (m + 1)$ [29,38,42]:

$$H_k(0) = \begin{pmatrix} \varphi_k & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \varphi_k \end{pmatrix}, \tag{16}$$

where φ_k is a coefficient that is a fairly large number, which is selected experimentally.

For the data set $\widetilde{x}_k^*(i), u^*(i)$, the correction matrix is calculated:

$$H_k(i) = H_k(i - 1) - \frac{H_k(i - 1) \cdot \widetilde{x}_k(i) \cdot \widetilde{x}_k^T(i) \cdot H_k(i - 1)}{1 + \widetilde{x}_k(i) \cdot \widetilde{x}_k^T(i) \cdot H_k(i - 1)} \tag{17}$$

and coefficient vector

$$\bar{b}_k(i) = \bar{b}_k(i - 1) + H_k(i) \cdot \tilde{x}_k(i) \cdot \left(u^*(i) - \bar{b}_k^T(i - 1) \cdot \tilde{x}_k(i) \right) - \lambda \cdot \bar{b}_k(i - 1), \quad (18)$$

where λ is the regularization parameter. The desired value of the vector \bar{b}_k is equal to $\bar{b}_k(N)$.

In the proposed modification, in contrast to [29], a regularization term $-\lambda \cdot \bar{b}_k(i - 1)$ is added to the coefficient correction process. This helps reduce overfitting and improve estimation robustness. The parameter λ must be carefully selected, since large values of λ can lead to underfitting, and too small ones can lead to insufficient consideration of regularization. The proposed method with regularization is especially useful when working with data prone to noise or with high multicollinearity between features.

The conducted research made it possible to formulate a theorem on the convergence of a modified parametric identification algorithm with regularization.

Theorem 1. *If the correction matrix $H_k(i)$ satisfies the condition of positive definiteness at each i -th step, and there is also a constant $\mu > 0$ such that $\lambda \leq \mu$, then the sequence $\{\bar{b}_k(i)\}$ converges to the true values of the coefficients b_k , which minimize the model estimate root mean square error.*

Proof of Theorem 1. It is assumed that the difference between the current and next iteration steps is described by expression (18). Using the properties of the adjustment matrix $H_k(i)$ and the regularization conditions $\|b_k(i) - b_k(i - 1)\| \leq \text{const} \cdot \left(u^*(i) - \bar{b}_k^T(i - 1) \cdot \tilde{x}_k(i) \right)$, the sequence $\bar{b}_k(i)$ converges to the true values of the coefficients b_k , since each iteration step reduces the model estimation error, taking regulation into account. Analyzing the iteration process dynamics and the adjustment matrix $H_k(i)$ properties, we can establish that the sequence $\{\bar{b}_k(i)\}$ converges to the true values of the coefficients b_k , minimizing the model error. \square

Thus, the theorem on the modified parametric identification algorithm convergence with regularization confirms that when the adjustment and regularization matrix positive definiteness conditions are met, the sequence of coefficients converges to the true model parameters, which completes the proof of the theorem.

To start the structural identification algorithm ψ_n operation, according to [29], a fuzzy model with the lowest criterion value is selected:

$$J_k = \frac{1}{N} \cdot \sum_{i=1}^N \left(\frac{u^*(i) - u_k(\bar{x}_k^*(i))}{u^*(i)} \right)^2. \quad (19)$$

From the available models, the one with the lowest criterion value is selected. If only one model is available, for example in the first iteration, this step is skipped. The resulting fuzzy model is then trained on the current data by systematically correcting the coefficients b_l^θ using a multi-step least squares method (Equations (17) and (18)) and the membership function parameters $d_{k1,l}^\theta, d_{k2,l}^\theta$ using the backpropagation algorithm [34], which aims to minimize the squared error:

$$E_k(i) = \frac{1}{2} \cdot e_k^2(i) = \frac{1}{2} \cdot \left(u^*(i) - u_k(\bar{d}_k, \bar{x}_k^*(i)) \right)^2 \quad (20)$$

using the gradient method according to the expression

$$d_k(\lambda + 1) = d_k(\lambda) - \Delta d_k(\lambda) \quad (21)$$

with working step $\Delta d_k(\lambda) = h_k \cdot \frac{\partial E_k}{\partial d_k}$, where h_k is the working step parameter.

The chain rule for determining the partial derivative according to [29] has the form:

$$\frac{\partial E_k}{\partial d_{kl}^\theta} = \frac{\partial E_k}{\partial u_k} \cdot \frac{\partial u_k}{\partial w_k^\theta} \cdot \frac{\partial w_k^\theta}{\partial X_{kl}^\theta} \cdot \frac{\partial X_{kl}^\theta}{\partial d_{kl}^\theta}, \quad (22)$$

where $\bar{\partial d_{kl}^\theta} = (d_{k1,l}^\theta, d_{k2,l}^\theta)$ is the bell-shaped membership function (Figure 3) parameters' two-element vector. Each component of the partial derivative (22) is determined according to (8)–(12). In this case, according to (20), it is defined as:

$$\frac{\partial E_k}{\partial u_k} = \frac{\partial E_k}{\partial u_k} \cdot \frac{\partial u_k}{\partial w_k^\theta} \cdot \frac{\partial w_k^\theta}{\partial X_{kl}^\theta} \cdot \frac{\partial X_{kl}^\theta}{\partial d_{kl}^\theta}, \quad (23)$$

the components of which are determined: $\frac{\partial E_k}{\partial u_k}$ according to (20), $\frac{\partial u_k}{\partial w_k^\theta}$ according to (7) and (10), $\frac{\partial w_k^\theta}{\partial X_{kl}^\theta}$ according to (9), and $\frac{\partial X_{kl}^\theta}{\partial d_{kl}^\theta}$ as the bell-shaped membership function derivative with respect to $d_{k1,l}^\theta$ or $d_{k2,l}^\theta$:

$$\frac{\partial E_k}{\partial u_k} = (u^* - u_k), \quad (24)$$

$$\frac{\partial u_k}{\partial w_k^\theta} = \alpha \cdot \frac{u_k^\theta \cdot \sum_{j=1}^n w_k^j - w_k^\theta \cdot \sum_{j=1}^n w_k^j \cdot u_k^j}{\sum_{j=1}^n w_k^j} + \frac{\partial \epsilon_k}{\partial w_k^\theta} = \alpha \cdot \frac{u_k^\theta - w_k^\theta \cdot u_k}{\sum_{j=1}^n w_k^j} + \frac{\partial \epsilon_k}{\partial w_k^\theta}, \quad (25)$$

Since according to (9) $w_k^\theta = X_{k1}^\theta(x_{k1}^\theta(i)) \cdot X_{k2}^\theta(x_{k2}^\theta(i)) \cdot \dots \cdot X_{km}^\theta(x_{km}^\theta(i))$, (25) is rewritten as:

$$\frac{\partial u_k}{\partial w_k^\theta} = \alpha \cdot \frac{u_k^\theta - w_k^\theta \cdot u_k}{\sum_{j=1}^n w_k^j} + \frac{\partial \epsilon_k}{\partial X_{kk}^\theta(x_{kk}^\theta)}, \quad (26)$$

$$\frac{\partial w_k^\theta}{\partial X_{kl}^\theta(x_l)} = \prod_{\substack{j=1, \\ j \neq l}}^m X_{kj}^\theta(x_{kj}^*), \quad (27)$$

$$\frac{\partial X_{kl}^\theta}{\partial d_{k1,l}^\theta} = \frac{2 \cdot (x_{kl}^* - d_{k1,l}^\theta)}{(d_{k2,l}^\theta)^2 \cdot \left(1 + \left(\frac{x_{kl}^* - d_{k1,l}^\theta}{d_{k2,l}^\theta}\right)^2\right)^2}, \quad \frac{\partial X_{kl}^\theta}{\partial d_{k2,l}^\theta} = \frac{2 \cdot (x_{kl}^* - d_{k1,l}^\theta)}{(d_{k2,l}^\theta)^3 \cdot \left(1 + \left(\frac{x_{kl}^* - d_{k1,l}^\theta}{d_{k2,l}^\theta}\right)^2\right)^2}. \quad (28)$$

Taking into account (24)–(27), (23) is presented as:

$$\frac{\partial X_{kl}^\theta}{\partial d_{k1,l}^\theta} = (u^* - u_k) \cdot \left(\alpha \cdot \frac{u_k^\theta - w_k^\theta \cdot u_k}{\sum_{j=1}^n w_k^j} + \frac{\partial \epsilon_k}{\partial X_{kk}^\theta(x_{kk}^\theta)} \right) \cdot \left(\prod_{\substack{j=1, \\ j \neq l}}^m X_{kj}^\theta(x_{kj}^*) \right) \cdot \left(\frac{2 \cdot (x_{kl}^* - d_{k1,l}^\theta)}{(d_{k2,l}^\theta)^2 \cdot \left(1 + \left(\frac{x_{kl}^* - d_{k1,l}^\theta}{d_{k2,l}^\theta}\right)^2\right)^2} \right), \quad (29)$$

or

$$\frac{\partial X_{kl}^\theta}{\partial d_{k2,l}^\theta} = (u^* - u_k) \cdot \left(\alpha \cdot \frac{u_k^\theta - w_k^\theta \cdot u_k}{\sum_{j=1}^n w_k^j} + \frac{\partial \epsilon_k}{\partial X_{kk}^\theta(x_{kk}^\theta)} \right) \cdot \left(\prod_{\substack{j=1, \\ j \neq l}}^m X_{kj}^\theta(x_{kj}^*) \right) \cdot \left(\frac{2 \cdot (x_{kl}^* - d_{k1,l}^\theta)}{(d_{k2,l}^\theta)^3 \cdot \left(1 + \left(\frac{x_{kl}^* - d_{k1,l}^\theta}{d_{k2,l}^\theta}\right)^2\right)^2} \right). \quad (30)$$

The resulting expressions (29) and (30) are modified in comparison with similar ones [29] by taking into account the noise term (error) $\epsilon_k(i)$ and the network training tuning

coefficient α , as well as the bell-shaped membership function rather than the use of a sigmoid one.

The physical meaning of the derivative $\frac{\partial \epsilon_k}{\partial X_{kk}^\theta(x_{kk}^\theta)}$ is to measure the sensitivity of the error ϵ_k to changes in the variable $X_{kk}^\theta(x_{kk}^\theta)$ in the context of the compensator model. This derivative shows how changing the $X_{kk}^\theta(x_{kk}^\theta)$ value, which can be a parameter or input value of the model, affects the error ϵ_k . If the $\frac{\partial \epsilon_k}{\partial X_{kk}^\theta(x_{kk}^\theta)}$ value is large, this indicates that small changes in $X_{kk}^\theta(x_{kk}^\theta)$ can lead to large changes in the model error ϵ_k , which may be important to optimize model parameters or analyze its sensitivity to certain input data. This can also help in understanding which factors or variables have the greatest impact on the NFA modified closed HTE ACS compensator model (Figure 2) accuracy and stability.

This modification makes it possible to improve the model accuracy and generalization ability, taking into account the data characteristics and the noise level in them, provide more flexible approximation, and allow the model to better adapt to different data types, which in turn contributes to more efficient training and prediction results.

2.3. Emulator Model

The emulator is an HTE simplified dynamic model, which can be described by a mathematical equation. This emulator is a real engine abstraction, taking into account the basic physical and technical characteristics, but not including all the details and complexities inherent in the real device. It is used to analyze the HTE operation and simulate in various modes and conditions, allowing researchers and engineers to test and optimize control systems and engine parameters without involving the large resources and time costs associated with real tests. Helicopter TE emulators can also be used for training and ensuring safety in the helicopter operational operation, allowing engine behavior analysis in various situations and flight conditions. Based on (1), the emulator is described by the equation:

$$\hat{y}(i+1) = g(\hat{y}(i), \dots, \hat{y}(i-r), x(i), \dots, x(i-s), u(i), \dots, u(i-q), d(i), \overline{c_{em}}), \quad (31)$$

where $\hat{y}(i+1)$ represents the output signal next value prediction at time $i+1$ according to the emulator model, and $g(\bullet)$ is the emulator function that takes into account the current state \hat{y} , disturbance vector $x(i)$, control $u(i)$, an additional parameter $g(i)$, and a tuning parameters $\overline{c_{em}}$ vector.

A tuning parameters vector in the context of an emulator means averaging by this vector all components over time or over the entire available data sample. For example, if it represents an emulator model settings vector (such as weights or coefficients), then $\overline{c_{em}}$ is a vector whose every component is the corresponding vector $\overline{c_{em}}$ component average. Thus, if $\overline{c_{em}} = (c_{em1}, c_{em2}, \dots, c_{em n})$, then

$$\overline{c_{em}} = \left(\frac{1}{N} \cdot \sum_{i=1}^N c_{em1}(i), \frac{1}{N} \cdot \sum_{i=1}^N c_{em2}(i), \dots, \frac{1}{N} \cdot \sum_{i=1}^N c_{em m}(i) \right). \quad (32)$$

Based on [29], expression (31) has known orders r, s, q , similar to (1). After formalizing the variable $\hat{y}(i), \dots, \hat{y}(i-r), x(i), \dots, x(i-s), u(i), \dots, u(i-q) = x_{em}(i) = \{x_{em1}(i) \dots x_{em m}(i)\}$, expression (31) is also represented in the form of a fuzzy Sugeno model:

$$\begin{aligned} & \text{if } x_{em1}(i), \text{ there is, } X_{em1}^\theta \\ & x_{em2}(i) \text{ there is } X_{em2}^\theta \dots x_{em m}(i) \text{ there is } X_{em m}^\theta \\ \text{then } \hat{y}^\theta(i+1) &= b_{em0}^\theta + b_{em1}^\theta \cdot x_{em1}(i) + \dots + b_{em m}^\theta \cdot x_{em m}(i) + \epsilon_{em}(i). \end{aligned} \quad (33)$$

The analytical expression of a fuzzy emulator, presented in a six-layer AFNN form, similar to (12), has the form:

$$\hat{y}(i+1) = \sum_{\theta=1}^n \beta_{em}^{\theta} \cdot \hat{y}^{\theta}(i+1) \text{ or } \hat{y}(i+1) = \overline{b_{em}^T} \cdot \widetilde{x_{em}}(i), \quad (34)$$

in which, similarly to (9) and (10), β_{em}^{θ} and w_{em}^{θ} are calculated as:

$$\beta_{em}^{\theta} = \alpha_{em} \cdot \frac{w_{em}^{\theta}(i)}{\sum_{\theta=1}^n w_{em}^{\theta}(i)}, \quad (35)$$

$$w_{em}^{\theta} = X_{em1}^{\theta}(x_{em1}^{\theta}(i)) \cdot X_{em2}^{\theta}(x_{em2}^{\theta}(i)) \cdot \dots \cdot X_{em m}^{\theta}(x_{em m}^{\theta}(i)). \quad (36)$$

The coefficient α_{em} in an emulator model is a tuning parameter or weight that is used to control the contribution of each parameter or characteristic to the model's calculations. The α_{em} parameter allows us to more flexibly take into account the significance of each element of input data or the influence of model parameters on its output. Thus, the coefficient α_{em} scales or adjusts the significance of $\frac{w_{em}^{\theta}(i)}{\sum_{\theta=1}^n w_{em}^{\theta}(i)}$ in the controller model. Similar to the α coefficient in NNs, the α_{em} parameter is tuned or determined based on expert knowledge or the emulator model training process to achieve optimal model performance or accuracy in modeling and control tasks.

The algorithm for identifying coefficients $b_{em}(i)$ is identical to (14)–(18), that is:

$$H_{em}(i) = H_{em}(i-1) - \frac{H_{em}(i-1) \cdot \widetilde{x_{em}}(i) \cdot \widetilde{x_{em}}^T(i) \cdot H_{em}(i-1)}{1 + \widetilde{x_{em}}(i) \cdot \widetilde{x_{em}}^T(i) \cdot H_{em}(i-1)}, \quad (37)$$

$$b_{em}(i) = b_{em}(i-1) + H_{em}(i) \cdot \widetilde{x_{em}}(i) \cdot (\hat{y}(i) - b_{em}^T(i-1) \cdot \widetilde{x_{em}}(i)) - \lambda_{em} \cdot b_{em}(i-1), \quad (38)$$

where λ_{em} is the regularization parameter, $\widetilde{x_{em}}(i) = \beta_{em}^1(i) \dots \beta_{em}^i(i), \beta_{em}^1(i) \cdot x_{em1}(i), \dots, \beta_k^1(i) \cdot x_{emm}(i) \dots \beta_{em}^i(i) \cdot x_{emm}(i)^T$ is the modified input vector, and $\overline{b_{em}} = (b_{em0}^1 \dots b_{em0}^n, b_{em1}^1 \dots b_{em1}^n, \dots, b_{emm}^1 \dots b_{emm}^n)^T$ is the adjustable coefficients vector.

For structural identification, as in [29], a similar criterion (19) is used, characterizing the average relative error:

$$J_{em} = \frac{1}{N+1} \cdot \sum_{i=1}^N \left(\left| \frac{y(i+1) - \hat{y}(i+1)}{y(i+1)} \right| \right). \quad (39)$$

The emulator bell-shaped membership function parameters $d_{em1,l}^{\theta}, d_{em2,l}^{\theta}$ are determined by the backpropagation method by minimizing the quadratic discrepancy similarly to (20) as:

$$E_{em}(i+1) = \frac{1}{2} \cdot e_{em}^2(i+1) = \frac{1}{2} \cdot \left(y(i+1) - \hat{y}(\overline{d_{em}}, \overline{x_{em}}(i)) \right)^2 \quad (40)$$

using the gradient method according to the expression

$$d_{em}(\lambda+1) = d_{em}(\lambda) - \Delta d_{em}(\lambda) \quad (41)$$

with working step $\Delta d_{em}(\lambda) = h_{em} \cdot \frac{\partial E_{em}}{\partial d_{em}}$, where h_{em} is the working step parameter.

The chain rule for determining the partial derivative according to [29], similar to (22), has the form:

$$\frac{\partial E_{em}}{\partial d_{em}} = \frac{\partial E_{em}}{\partial y} \cdot \frac{\partial y}{\partial w_{em}^{\theta}} \cdot \frac{\partial w_{em}^{\theta}}{\partial X_{em l}^{\theta}} \cdot \frac{\partial X_{em l}^{\theta}}{\partial d_{em l}^{\theta}}, \quad (42)$$

which, after mathematical transformations similar to (24)–(28), is represented as:

$$\frac{\partial E_{em}}{\partial \bar{d}_{em}} = (y - \hat{y}) \cdot \left(\alpha_{em} \cdot y \cdot \frac{1 - w_{em}^\theta}{\sum_{j=1}^n w_{em}^j} + \frac{\partial \epsilon_{em}}{\partial X_{em\ k}^\theta (x_{em\ k}^\theta)} \right) \cdot \left(\prod_{\substack{j=1, \\ j \neq l}}^m X_{em\ j}^\theta (x_{em\ j}^*) \right) \cdot \left(\frac{2 \cdot (x_{em\ l}^* - d_{em1,l}^\theta)}{(d_{em2,l}^\theta)^2 \cdot \left(1 + \left(\frac{x_{em\ l}^* - d_{em1,l}^\theta}{d_{em2,l}^\theta} \right)^2 \right)^2} \right), \quad (43)$$

or

$$\frac{\partial E_{em}}{\partial \bar{d}_{em}} = (y - \hat{y}) \cdot \left(\alpha_{em} \cdot y \cdot \frac{1 - w_{em}^\theta}{\sum_{j=1}^n w_{em}^j} + \frac{\partial \epsilon_{em}}{\partial X_{em\ k}^\theta (x_{em\ k}^\theta)} \right) \cdot \left(\prod_{\substack{j=1, \\ j \neq l}}^m X_{em\ j}^\theta (x_{em\ j}^*) \right) \cdot \left(\frac{2 \cdot (x_{em\ l}^* - d_{em1,l}^\theta)}{(d_{em2,l}^\theta)^3 \cdot \left(1 + \left(\frac{x_{em\ l}^* - d_{em1,l}^\theta}{d_{em2,l}^\theta} \right)^2 \right)^2} \right). \quad (44)$$

Parametric and structural identifications stop when a condition similar to (13) is met:

$$J_{em} = \frac{1}{N} \cdot \sum_{i=0}^N \left| \frac{y(i+1) - \hat{y}(i+1)}{y(i+1)} \right| \leq J_{em\ nom}, \quad (45)$$

where J_{em} is the emulator average relative error with an acceptable nominal value of the relative error $J_{em\ nom}$.

2.4. Controller Model

In an NFA modified closed onboard HTE ACS (Figure 2) with a regulator, emulator, and compensator, the regulator is a system part that takes as input the system current state and issues a control action to correct this state in accordance with the desired requirements or control aims. The controller uses fuzzy rules and a knowledge base to make control decisions based on current historical data and input variables.

To construct a controller model according to [29], it is assumed that the HTE described by (1) is reversible, that is, there is a controller model function $f_p(\bullet)$ of the form:

$$u_{reg}(i) = f(\hat{y}(i), \dots, \hat{y}(i - r + 1), x(i), \dots, x(i - s + 1), u_{reg}(i), \dots, u_{reg}(i - q + 1), y_{nom}, d(i)), \quad (46)$$

in which the values $i - r + 1, i - s + 1, i - q + 1$ indicate specific points in time in the past (relative to the i -th point in time), which are taken into account when determining the variables that the controller model $u_{reg}(i)$ influence. These values are used to evaluate the system state and input variables in the past in order to make a decision on the necessary control action $u_{reg}(i)$ at the current i -th time. The variable y_{nom} represents the system state nominal value y , which is likely used in the controller model to compare with the system current state and calculate the corrective control action.

After formalizing the variable $\hat{y}(i), \dots, \hat{y}(i - r + 1), x(i), \dots, x(i - s + 1), u_{reg}(i), \dots, u_{reg}(i - q + 1) = x_{reg}(i) = \{x_{reg1}(i) \dots x_{regm}(i)\}$ expression (46) is also represented as a fuzzy Sugeno model of the form:

$$\begin{aligned} &\text{if } x_{reg1}(i), \text{ there is, } X_{reg1}^\theta \\ &x_{reg2}(i) \text{ there is } X_{reg2}^\theta \dots x_{reg\ t}(i) \text{ there is } X_{reg\ t}^\theta \\ &\text{then } u_{reg}(i) = b_{reg0}^\theta + b_{reg1}^\theta \cdot x_{reg1}(i) + \dots + b_{reg\ t}^\theta \cdot x_{reg\ t}(i) + \epsilon_{reg}(i). \end{aligned} \quad (47)$$

The fuzzy controller analytical expression is similar to (12), taking into account [29], and has the form:

$$u_{reg}(i) = \frac{\sum_{\theta=1}^n w_{reg}^\theta(i) \cdot u_{reg}^\theta(i)}{\sum_{\theta=1}^n w_{reg}^\theta(i)} \text{ or } u_{reg}(i) = \overline{b_{reg}^T} \cdot \tilde{x}_{reg}(i), \quad (48)$$

which is a control actions weighted average from all controller components $u_{reg}^\theta(i)$ multiplied by its weight $w_{reg}^\theta(i)$ and then divided by the all-weights sum $\sum_{\theta=1}^n w_{reg}^\theta(i)$. This makes it possible to take into account the contribution of each component in the overall control action formation. The coefficient α_{reg} is multiplied by this weighted average result, which serves as an adjustment or the controller output control action $u_{reg}(i)$ scaling. In (48), similarly to (9) and (10), β_{em}^θ and w_{em}^θ are calculated as:

$$\beta_{reg}^\theta = \alpha_{reg} \cdot \frac{w_{reg}^\theta(i)}{\sum_{\theta=1}^n w_{reg}^\theta(i)}, \tag{49}$$

$$w_{reg}^\theta = X_{reg1}^\theta(x_{reg1}^\theta(i)) \cdot X_{reg2}^\theta(x_{reg2}^\theta(i)) \cdot \dots \cdot X_{regt}^\theta(x_{regt}^\theta(i)), \tag{50}$$

where $\tilde{x}_{reg}(i) = \beta_{reg}^1(i) \dots \beta_{reg}^t(i), \beta_{reg}^1(i) \cdot x_{reg1}(i), \dots, \beta_{reg}^1(i) \cdot x_{regt}(i) \dots \beta_{reg}^t(i) \cdot x_{regt}(i)^T$ is a modified input vector.

According to [29], inverse controller model structure and parameters identification is carried out in two stages. At the first stage, we use the emulator model and a one-dimensional search algorithm at the points $u^*(i) = u_{reg}^*(i) + u_k(i)$ and, accordingly, the regulatory influences $u_{reg}^*(i)$, at which the emulator error $e_{em}(i + 1) = |y(i + 1) - \hat{y}(i + 1)|$ satisfies the constraints $\delta y_{min} \leq e_{em}(i + 1) \leq \overline{\delta y_{min}}, i = 0 \dots N - 1$.

At the second stage, structural identification and linear equations coefficients identification $\overline{b_{reg}}$ and the bell-shaped membership functions $\overline{d_{reg}}$ parameters are carried out. The coefficients vector $\overline{b_{reg}}$ is calculated using the recurrent least squares method identical to (14)–(18), that is:

$$H_{reg}(i) = H_{reg}(i - 1) - \frac{H_{reg}(i - 1) \cdot \tilde{x}_{reg}(i) \cdot \tilde{x}_{reg}^T(i) \cdot H_{reg}(i - 1)}{1 + \tilde{x}_{reg}(i) \cdot \tilde{x}_{reg}^T(i) \cdot H_{reg}(i - 1)}, \tag{51}$$

$$b_{reg}(i) = b_{reg}(i - 1) + H_{reg}(i) \cdot \tilde{x}_{reg}(i) \cdot (u_{reg}^*(i) - b_{reg}^T(i - 1) \cdot \tilde{x}_{reg}(i)) - \lambda_{reg} \cdot b_{reg}(i - 1), \tag{52}$$

where λ_{reg} is the regularization parameter, $\overline{b_{reg}} = (b_{reg0}^1 \dots b_{reg0}^n, b_{reg1}^1 \dots b_{reg1}^n, \dots, b_{regt}^1 \dots b_{regt}^n)^T$ is the vector adjustable coefficients, and $H_{reg}(0) = \gamma \cdot I, \gamma \gg 1$.

For structural identification, as in [29], a similar criterion (19) is used, characterizing the average relative error:

$$J_{reg} = \frac{1}{N} \cdot \sum_{i=0}^N \left(\left| \frac{u_{reg}^*(i) - u_{reg}(i)}{u_{reg}^*(i)} \right| \right). \tag{53}$$

Structural and parametric identification is completed when a condition similar to (13) is met:

$$J_{reg} = \frac{1}{N} \cdot \sum_{i=0}^N \left| \frac{u_{reg}^*(i) - u_{reg}(i)}{u_{reg}^*(i)} \right| \leq J_{reg\ nom}, \tag{54}$$

where J_{reg} is the emulator average relative error with the permissible nominal value of the relative error $J_{reg\ nom}$.

The emulator bell-shaped membership function parameters $d_{reg1,l}^\theta, d_{reg2,l}^\theta$ are determined by the backpropagation method by minimizing the quadratic discrepancy similarly to (20) as:

$$E_{reg}(i + 1) = \frac{1}{2} \cdot e_{reg}^2(i + 1) = \frac{1}{2} \cdot (y_{nom} - \hat{y}(\overline{d_{reg}}, \overline{x_{reg}}(i)))^2 \tag{55}$$

using the gradient method according to the expression

$$d_{reg}(\lambda + 1) = d_{reg}(\lambda) - \Delta d_{reg}(\lambda) \tag{56}$$

with working step $\Delta d_{reg}(\lambda) = h_{reg} \cdot \frac{\partial E_{reg}}{\partial d_{reg}}$, where h_{reg} is the working step parameter.

The chain rule for determining the partial derivative according to [29], similar to (22), has the form:

$$\frac{\partial E_{reg}}{\partial d_{reg}} = \frac{\partial E_{reg}}{\partial u_{reg}} \cdot \frac{\partial u_{reg}}{\partial w_{reg}^\theta} \cdot \frac{\partial w_{reg}^\theta}{\partial X_{reg}^\theta} \cdot \frac{\partial X_{reg}^\theta}{\partial d_{reg}}, \tag{57}$$

which, after mathematical transformations similar to (24)–(28), is represented as:

$$\frac{\partial E_{reg}}{\partial d_{reg}} = (u_{reg}^*(i) - u_{reg}(i)) \cdot \left(\alpha_{reg} \cdot \frac{u_{reg}^\theta(i) - w_{reg}^\theta \cdot u_{reg}(i)}{\sum_{j=1}^n w_{reg}^j} + \frac{\partial \epsilon_{reg}}{\partial X_{reg}^\theta(x_{reg}^\theta)} \right) \cdot \left(\prod_{\substack{j=1, \\ j \neq l}}^n X_{reg}^\theta(x_{reg}^* j) \right) \cdot \left(\frac{2 \cdot (x_{reg}^* l - d_{reg1,l}^\theta)}{(d_{reg2,l}^\theta)^2 \cdot \left(1 + \left(\frac{x_{reg}^* l - d_{reg1,l}^\theta}{d_{reg2,l}^\theta} \right)^2 \right)^2} \right), \tag{58}$$

or

$$\frac{\partial E_{em}}{\partial d_{em}} = (u_{reg}^*(i) - u_{reg}(i)) \cdot \left(\alpha_{reg} \cdot \frac{u_{reg}^\theta(i) - w_{reg}^\theta \cdot u_{reg}(i)}{\sum_{j=1}^n w_{reg}^j} + \frac{\partial \epsilon_{reg}}{\partial X_{reg}^\theta(x_{reg}^\theta)} \right) \cdot \left(\prod_{\substack{j=1, \\ j \neq l}}^n X_{reg}^\theta(x_{reg}^* j) \right) \cdot \left(\frac{2 \cdot (x_{reg}^* l - d_{reg1,l}^\theta)}{(d_{reg2,l}^\theta)^3 \cdot \left(1 + \left(\frac{x_{reg}^* l - d_{reg1,l}^\theta}{d_{reg2,l}^\theta} \right)^2 \right)^2} \right). \tag{59}$$

2.5. Clarifying the Emulator Model Taking into Account the Controller Model

Taking u_k constant, and taking into account that $u_{reg} = x_{em1}$, i.e., $du_{reg} = d(u_k + u_{reg}) = dx_{em1}$ according to [29], the object analytical Jacobian expression is defined $\frac{\partial \hat{y}(i+1)}{\partial x_{em1}}$, which is the sum of derivatives of the products of two functions β^θ and \hat{y}^θ :

$$\frac{\partial \hat{y}_l}{\partial x_{em1}} = \sum_{\theta=1}^n \left(y^\theta \cdot \frac{\partial \beta^\theta}{\partial x_{em1}} + \beta^\theta \cdot \frac{\partial y^\theta}{\partial x_{em1}} \right). \tag{60}$$

Based on [29] in accordance with the chain rule, the first term total derivative has the form:

$$y^\theta \cdot \frac{\partial \beta^\theta}{\partial x_{em1}} = y^\theta \cdot \left(\frac{\partial \beta_{em}^\theta}{\partial w_{em}^\theta} \cdot \frac{\partial w_{em}^\theta}{\partial X_{em}^\theta} \cdot \frac{\partial X_{em}^\theta}{\partial x_{em1}} \right), \tag{61}$$

where

$$y^\theta \cdot \frac{\partial \beta_{em}^\theta}{\partial x_{em1}} = \frac{y^\theta - w_{em}^\theta \cdot \hat{y}}{\sum_{j=1}^n w_{em}^j}, \quad \frac{\partial w_{em}^\theta}{\partial x_{em1}} = \prod_{l=2}^m X_{em}^\theta(x_{em1}),$$

$$\frac{\partial \beta_{em}^\theta}{\partial w_{em}^\theta} = \alpha \cdot \frac{\beta_{em}^\theta - w_{em}^\theta \cdot \beta_{em}}{\sum_{j=1}^n w_{em}^j} + \frac{\partial \epsilon_k}{\partial X_{em}^\theta(x_{em}^\theta)}, \quad \frac{\partial X_{em}^\theta(x_{em1})}{\partial x_{em1}} = \frac{2 \cdot (x_{em1} - d_{em1,l}^\theta)}{(d_{em2,l}^\theta)^2 \cdot \left(1 + \left(\frac{x_{em1} - d_{em1,l}^\theta}{d_{em2,l}^\theta} \right)^2 \right)^2}, \quad \beta^\theta \cdot \frac{\partial y^\theta}{\partial x_{em1}} = b_{em1}^\theta \cdot \beta_{em}^\theta. \tag{62}$$

Taking into account (62), expression (61) takes the form:

$$\frac{\partial \hat{y}_l}{\partial x_{em1}} = \sum_{\theta=1}^n \left(y^\theta \cdot \left(\alpha \cdot \frac{\beta_{em}^\theta - w_{em}^\theta \cdot \beta_{em}}{\sum_{j=1}^n w_{em}^j} + \frac{\partial \epsilon_k}{\partial X_{em}^\theta(x_{em}^\theta)} \cdot \prod_{l=2}^m X_{em}^\theta(x_{em1}) \right) \cdot \frac{2 \cdot (x_{em1} - d_{em1,l}^\theta)}{(d_{em2,l}^\theta)^2 \cdot \left(1 + \left(\frac{x_{em1} - d_{em1,l}^\theta}{d_{em2,l}^\theta} \right)^2 \right)^2} + \beta^\theta \cdot y^\theta \cdot \left(\frac{y^\theta - w_{em}^\theta \cdot \hat{y}}{\sum_{j=1}^n w_{em}^j} \cdot \prod_{l=2}^m X_{em}^\theta(x_{em1}) \cdot \frac{2 \cdot (x_{em1} - d_{em1,l}^\theta)}{(d_{em2,l}^\theta)^2 \cdot \left(1 + \left(\frac{x_{em1} - d_{em1,l}^\theta}{d_{em2,l}^\theta} \right)^2 \right)^2} \right) \right). \tag{63}$$

The advantage of (63) over [29], where a similar expression is represented as $\frac{\partial \hat{y}_l}{\partial x_{em1}} = \sum_{\theta=1}^n \frac{y^\theta - w_{em}^\theta \cdot \hat{y}}{\sum_{j=1}^n w_{em}^j} \cdot \prod_{l=2}^m X_{em}^\theta(x_{em1}) \cdot d_{em1,l}^\theta \cdot (1 - X_{em}^\theta(x_{em1})) + b_{em1}^\theta \cdot \beta_{em}^\theta$, consists in its ability to

take into account multidimensional interactions and nonlinear effects between variables through complex weighting coefficients and derivatives, which makes it possible to more accurately model dependencies in a multidimensional NFA modified closed HTE ACS (Figure 2).

2.6. Six-Layer NFN AFNN Training Algorithm

At the training initial stage, a parameter based on a six-layer NFN AFNN, $x_k(i)$ is accepted as the k -th input parameter for the i -th sample in the training set. The fuzzy transformation in the fuzzy layer uses the membership functions $\mu_{kj}(x_k(i))$ to transform $x_k(i)$ into the fuzzy value $X_{kj}^\theta(i)$, that is, $X_{kj}^\theta(i) = \mu_{kj}(x_k(i))$.

For each θ -th rule and each premise k , the premises $w_k^\theta(i)$ truth degrees are calculated as:

$$w_k^\theta(i) = X_{k1}^\theta(i) \times X_{k2}^\theta(i) \times \dots \times X_{km}^\theta(i). \quad (64)$$

Next, for each rule of the θ -th rule and each premise k , the relative normalized value is calculated, taking into account the parameter α , as:

$$\beta_k^\theta(i) = \alpha \cdot \frac{w_k^\theta(i)}{\sum_{j=1}^n w_k^j(i)}. \quad (65)$$

The next training step is to define a loss function L that measures the difference between the actual output $y(i)$ and the model predicted output $\hat{y}(i)$:

$$L = \sum_{i=1}^N (y(i) - \hat{y}(i))^2 + \vartheta \cdot t \cdot \sum_{i=1}^N (y(i) - \hat{y}(i))^2 + \lambda \cdot \sum_{k=1}^n \sum_{\theta} \theta^2, \quad (66)$$

where ϑ is the coefficient that regulates the influence of the adaptive multiplier, and t is the current training epoch number, the introduction of which into (66) allows us to adapt the loss function during the training process contribution. By adding the term $\lambda \cdot \|\theta\|^2$ to (66), in which λ is the regularization coefficient and $\|\theta\|^2$ is the L2-norm of the model parameters θ , it is possible to model the complexity control by penalizing large parameter values. The regularization coefficient λ determines the regularization strength: large values of λ increase the penalty for large parameters, which can help prevent overfitting.

For each θ -th model parameter, its gradient is calculated using the chain rule and gradient descent as:

$$\frac{\partial L}{\partial \theta} = 2 \cdot \lambda \cdot \theta + \sum_{i=1}^N \frac{\partial L}{\partial \hat{y}(i)} \cdot \frac{\partial \hat{y}(i)}{\partial \theta}. \quad (67)$$

Using gradients, the model parameters θ are updated using the gradient descent method:

$$\theta_{t+1} = \theta_t - \eta_t \cdot \left(\frac{\partial L}{\partial \theta} + 2 \cdot \lambda \cdot \theta_t \right). \quad (68)$$

where η_t is the current adaptive training rate at the t -th step. The training rate becomes adaptive by applying algorithms such as Adam or RMSprop [42,43], which automatically adapt the learning rate based on the gradient history.

Thus, adding adaptive elements to the training algorithm for a six-layer NFN AFNN will allow the HTE ACS to change its parameters and settings in real time based on changing conditions or external influences. The proposed algorithm can use feedback data to automatically adjust system parameters to achieve optimal performance even under variable conditions.

2.7. Observer Block Introduction into an NFA Modified Closed Onboard TE ACS

For a modified nonlinear difference system (1), where only the output measurements $y(i)$ are available, and the state $x(i)$ is not available for direct measurement, to estimate the

system state $x(i)$ based on the available measurements of $y(i)$ and control actions $u(i)$, the use of an observer block is proposed. It is assumed that the system state $x(i)$ evolves in accordance with the discrete dynamic equation:

$$x(i+1) = g(x(i), u(i), v(i)), \quad (69)$$

where $v(i)$ is a noise or disturbances vector, and the function g describes the changes dynamics in the system state based on the current state $x(i)$, control $u(i)$, and disturbance $v(i)$.

The observer block aim is to estimate a state $\hat{x}(i)$ that is approximately equal to the real state $x(i)$ using available measurements $y(i)$ and control inputs $u(i)$.

One observer method for discrete systems that can have significant advantages over the Extended Kalman Filter (EKF) is the Matching State Observer algorithm—the Innovation Matching Observer algorithm [44]. The state-matching observer algorithm advantage over the extended Kalman filter is its simplicity and the absence of the need to linearize nonlinear measurement functions, making it more efficient and robust for nonlinear systems.

The basic idea of the Matching State Observer method is to use measurements of $y(i)$ to directly determine $x(i)$, minimizing estimation error. This method's advantage is its simplicity and efficiency, especially in cases where the measurements are directly dependent on the state of the system.

According to the Matching State Observer method, the system state prediction is defined as:

$$\hat{x}(i+1|i) = A \cdot \hat{x}(i|i) + B \cdot u(i), \quad (70)$$

where $\hat{x}(i|i)$ is the system state current assessment, and A and B are the system or control matrices.

Measurement-based state correction (update) is defined as:

$$y(i) : \hat{x}(i+1|i+1) = \hat{x}(i+1|i) + K(i+1) \cdot (y(i+1) - C \cdot \hat{x}(i+1|i)), \quad (71)$$

where C is the measurement matrix that the system state relates to measurements $y(i)$, and $K(i+1)$ is the observer gain matrix, which is chosen to the estimation error minimize, which is defined as:

$$K(i+1) = P(i+1) \cdot C^T \cdot (C \cdot P(i+1) \cdot C^T + R)^{-1}, \quad (72)$$

where $P(i+1)$ is the state covariance error estimate, and R is the measurement noise covariance matrix.

This observer method directly uses measurements of $y(i)$ to adjust the state estimate $\hat{x}(i+1|i)$, which can be an effective way to estimate state in discrete systems, especially if the measurements have a direct and reliable relation with the system state. It also has a simple implementation and does not require the nonlinear measurement function linearization, which can be an advantage over the extended Kalman filter in certain scenarios.

3. Results

3.1. Input Data Analysis and Pre-Processing

To test the developed NFA modified closed onboard HTE ACS, including a controller, compensator, emulator, and observer unit, implemented in the six-layer NFN AFNN form, a computational experiment was carried out, for which a personal computer with a processor was used, i.e., AMD Ryzen 5 5600, 32 KB L3 cache, Zen 3 architecture, six cores, 12 threads, 3.5 GHz, RAM—32 GB DDR-4.

The data used to train the neural network include the HTE thermodynamic characteristics collected during flights, such as gas-generator rotor r.p.m. n_{TC} , free turbine rotor speed n_{FT} , and gas temperature in front of the compressor turbine T_G^* . The developed NFA modified closed onboard HTE ACS initial data were obtained from experimental research conducted on the TV3-117 engine (Table 1) [26,27,45–47].

Table 1. Initial data fragment for the developed NFA modified closed onboard TE ACS.

Number	TE Thermogas-Dynamic Parameter Absolute Value		
	n_{TC}	n_{FT}	T_G^*
1	0.943	0.929	0.932
2	0.982	0.933	0.964
...
22	0.973	0.919	0.970
...
38	0.975	0.920	0.971
...
54	0.984	0.931	0.979
...
112	0.962	0.907	0.953
...
188	0.970	0.913	0.966
...
256	0.981	0.973	0.953

The input data (the training set) pre-processing is given in [26,27,45–47]. As part of the input data preliminary processing, it was proven that the training set is homogeneous according to the Fisher–Pearson [48] and Fisher–Snedecor [49] criteria, whose values were 3.588 and 1.28, respectively. The obtained values of these statistical criteria are less than their critical values (22.362 and 3.44, respectively), with 13 free powers and a 0.01 significance level. According to [26,27,45–47], a control sample (67%) and a test sample (33%) were separated from the training sample (Table 2), and this representativeness was confirmed by cluster analysis [50,51] using the k -means method, during which the input data values were divided into eight classes and the metric distance between them was calculated as $C_i = \operatorname{argmin}_j \|x_i - \mu_j\|^2$, in which μ_j are the initial centroids and $\|x_i - \mu_j\|^2$ is the Euclidean distance between x_i and μ_j (Figure 5). In this case, the μ_j values are recalculated as $\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$, which is the number of objects in the j -th cluster. Calculations of C_i and μ_j continued until changes in the cluster distribution became negligible.

Table 2. Results of forming training, control, and test samples (author’s research).

Sample	Description	Sufficient Sample Size
Training	It was used to train a model.	256 elements (100%)
Control	It was used to evaluate the model.	172 elements (67% of the training sample)
Test	It was used for final model verification.	84 elements (33% of the training sample)

Similar research was conducted, during which the following were separated from the training sample (Table 2) and identified: a control sample (60%) and a test sample (40%); a control sample (50%) and a test sample (50%); a control sample (70%) and a test sample (30%). The obtained metric distance values with such combinations are 1.75...3.38 times greater than with the control sample (67%) and a test sample (33%). This indicates the optimal choice of control and test sample size.

Thus, the training sample elements’ preprocessing results made it possible to form the optimal training, control, and test sample sizes (Table 2).

3.2. The Six-Layer NFN AFNN Training Results

In the NFN AFNN training process with the proposed algorithm (64)–(68), the values of the accuracy (Accuracy) (Figure 6) and loss (Loss) (Figure 7) functions were estimated for the training and validation samples; the blue curve indicates training results on the training set, and the orange curve indicates validation results on the control set.

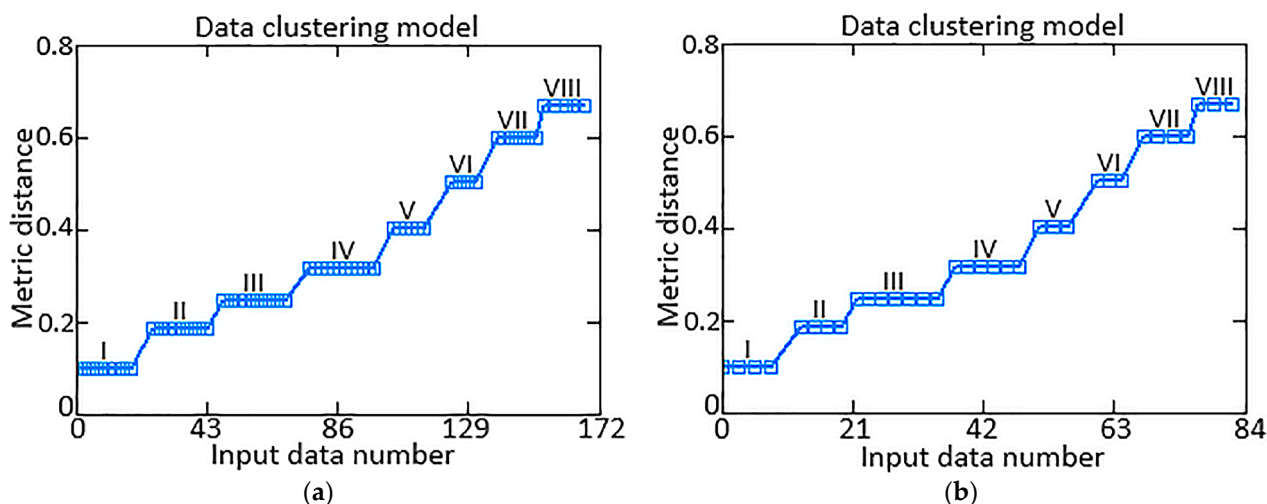


Figure 5. Cluster analysis results: (a) original experimental dataset (I..VIII-classes); (b) training dataset.

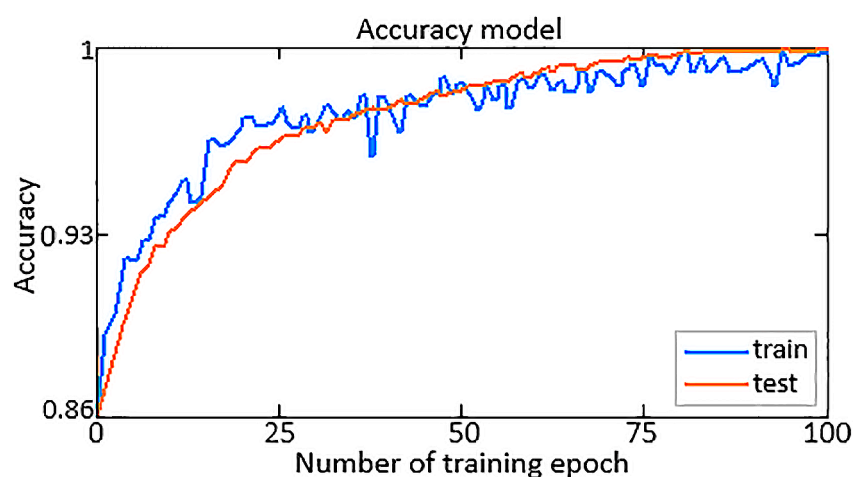


Figure 6. Diagram of changes in the neural network accuracy function with 1000 iterations.

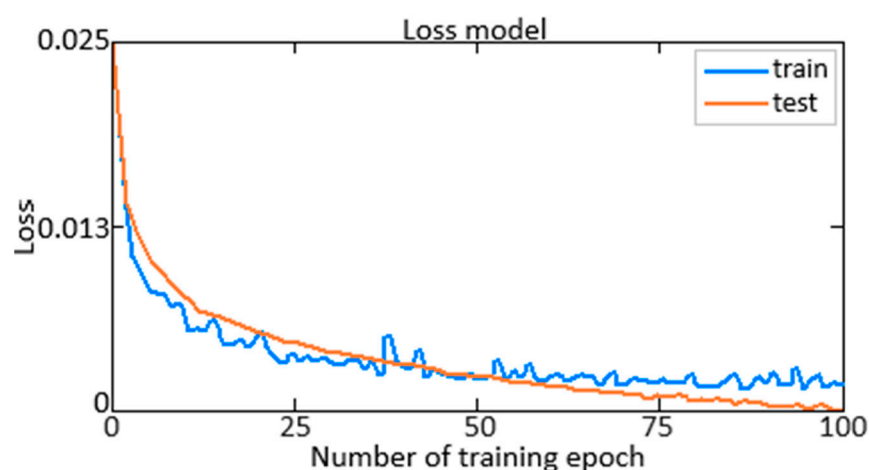


Figure 7. Diagram of changes in the neural network loss with 1000 iterations.

The accuracy metric reflects the model effectiveness in determining the neuro-fuzzy HTE ACS parameters, showing the percentage of correctly calculated parameters. In turn, the loss function demonstrates the degree to which the model's predictions deviate from the actual values, which helps optimize the training process by reducing this loss. By analyzing

the resulting diagrams of the accuracy (Accuracy) (Figure 6) and loss (Loss) (Figure 7) functions during the training process, conclusions are drawn about how effectively the model is trained, and measures are taken to improve its performance.

According to Figure 6, almost maximum accuracy has been achieved, and according to Figure 7, the loss value does not exceed 0.025, which indicates the model's high training efficiency on available data and its ability to accurately generalize to new data.

It is worth noting that for a bell-shaped membership function with 60 iterations, the value of the accuracy metric reaches 0.9998 (99.98%), and the value of the loss function does not exceed 0.0002 (0.2%), which indicates the high efficiency and accuracy of the AFNN network when using the bell-shaped function accessories. The results obtained indicate the successful training of the neuro-fuzzy AFNN network, with good convergence and minimal error, which emphasizes the importance of correct membership function choice in the design process of a neuro-fuzzy HTE ACS. At the same time, the loss values in both samples do not exceed 0.025 (or 2.5%), which indicates a low level of error in determining the neuro-fuzzy HTE ACS parameters. Such results indicate the successful training of the AFNN network with the proposed algorithm and its ability to provide the neuro-fuzzy HTE ACS accurate parameters with a high degree of confidence.

Determining confidence in the context of machine learning and neural network models typically involves assessing likelihood or the model's confidence in its predictions. Depending on the type of model (e.g., classification or regression), the confidence degree can be expressed differently [52]. This work uses a method for assessing the AFNN network's confidence which consists of using maximum probability. For classification problems, a softmax function is often used to determine the membership probabilities in each class. After model training, having received the prediction \hat{y} for input data x , we can obtain the probability $p(y = k|x)$ belonging to the k -th class using softmax:

$$p(y = k|x) = \frac{\exp(\hat{y}_k)}{\sum_{j=1}^K (\exp(\hat{y}_j))}, \quad (73)$$

where \hat{y} is the model outputs vector before applying softmax for input x , \hat{y}_k is the vector \hat{y} k -th element, and K is the class number.

Figure 8 shows a diagram of the maximum probability (Max Probability Estimation) for the AFNN network "degree of confidence", which shows how the AFNN network confidence in obtaining the neuro-fuzzy HTE ACS parameters is distributed: (a) is the n_{TC} parameter, (b) is the n_{FT} parameter, and (c) is the T_G^* parameter.

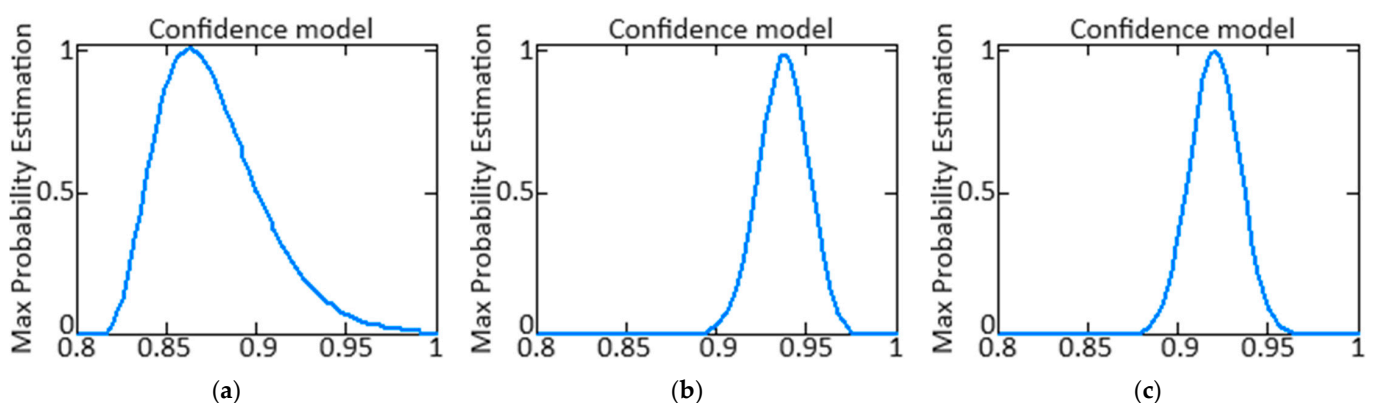


Figure 8. Maximum probability estimation diagram for "degree of confidence": (a) is the n_{TC} parameter, (b) is the n_{FT} parameter, and (c) is the T_G^* parameter.

Table 3 presents the comparative analysis results of the accuracy metric and the loss function calculations when using various membership functions in the AFNN network training process using the proposed algorithm (64)–(68).

Table 3. The different membership functions comparison results.

Membership Function Name	Max. Accuracy	Max. Loss	Training Epochs Optimal Number	Training Time, Seconds
Triangular	0.9952	0.0048	100	7.320
Trapezoidal	0.9465	0.0535	220	12.832
Symmetrical Gaussian	0.9779	0.0221	80	5.673
Two-sided Gaussian	0.9786	0.0214	80	5.918
Difference of two sigmoids	0.9501	0.0499	140	9.752
Product of two sigmoid	0.9515	0.0485	140	9.776
Generalized bell-shaped	0.9998	0.0002	60	4.061

In the process of training the NFN AFNN, similarly to [28,29], the bell-shaped membership function (Table 4) parameters were updated using an adaptive training rate according to the expressions:

$$d_{k1,l}^\theta(i+1) = d_{k1,l}^\theta(i) - \eta_t \cdot \frac{\partial E_k}{\partial d_{k1,l}^\theta}, \quad d_{k2,l}^\theta(i+1) = d_{k2,l}^\theta(i) - \eta_t \cdot \frac{\partial E_k}{\partial d_{k2,l}^\theta}, \quad (74)$$

where η_t is the current adaptive training rate at the t -th step.

Table 4. The bell-shaped membership function parameters values before and after training of AFNN neural fuzzy network (author’s development, based on [28]).

Step	Parameters	Values					
		ANFIS	AFNN	ANFIS	AFNN	ANFIS	AFNN
Before training the NFN	$d_{k1,l}^\theta$	0.100	0.10	0.100	0.100	0.100	0.100
	$d_{k2,l}^\theta$	0.250	0.25	0.500	0.500	0.750	0.750
After training the NFN	$d_{k1,l}^\theta$	0.120	0.117	0.110	0.112	0.130	0.118
	$d_{k2,l}^\theta$	0.290	0.263	0.530	0.508	0.760	0.751

The quality of the bell-shaped function parameters calculating $d_{k1,l}^\theta$ and $d_{k2,l}^\theta$ according to (74) using the adaptive training rate in comparison with the expressions $d_{k1,l}^\theta(i+1) = d_{k1,l}^\theta(i) - \eta \cdot \frac{\partial E_k}{\partial d_{k1,l}^\theta}$, $d_{k2,l}^\theta(i+1) = d_{k2,l}^\theta(i) - \eta \cdot \frac{\partial E_k}{\partial d_{k2,l}^\theta}$, applied in [28], was estimated by the values of RMSE (square root of root mean square error), MAE (mean absolute error), and MAPE (average absolute percentage error):

$$RMSE_{d_{k1,l}^\theta} = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n (d_{k1,l,i}^\theta - \hat{d}_{k1,l,i}^\theta)^2}, \quad RMSE_{d_{k2,l}^\theta} = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n (d_{k2,l,i}^\theta - \hat{d}_{k2,l,i}^\theta)^2}, \quad (75)$$

$$MAE_{d_{k1,l}^\theta} = \frac{1}{n} \cdot \sum_{i=1}^n |d_{k1,l,i}^\theta - \hat{d}_{k1,l,i}^\theta|, \quad MAE_{d_{k2,l}^\theta} = \frac{1}{n} \cdot \sum_{i=1}^n |d_{k2,l,i}^\theta - \hat{d}_{k2,l,i}^\theta|, \quad (76)$$

$$MAPE_{d_{k1,l}^\theta} = \frac{100}{n} \cdot \sum_{i=1}^n \left| \frac{\hat{d}_{k1,l,i}^\theta - d_{k1,l,i}^\theta}{d_{k1,l,i}^\theta} \right|, \quad MAPE_{d_{k2,l}^\theta} = \frac{100}{n} \cdot \sum_{i=1}^n \left| \frac{\hat{d}_{k2,l,i}^\theta - d_{k2,l,i}^\theta}{d_{k2,l,i}^\theta} \right|, \quad (77)$$

where n is the number of obtained values $d_{k1,l}^\theta$ and $d_{k2,l}^\theta$, $d_{k1,l,i}^\theta$ and $d_{k2,l,i}^\theta$ are actual values of i -th parameters $d_{k1,l}^\theta$ and $d_{k2,l}^\theta$, and $\hat{d}_{k1,l,i}^\theta$ and $\hat{d}_{k2,l,i}^\theta$ are the calculated NFN (AFNN in this work, ANFIS in [28]) values of i -th parameters $d_{k1,l}^\theta$ and $d_{k2,l}^\theta$.

Table 5 shows the calculated RMSE, MAE, and MAPE value results for the bell-shaped membership function parameters $d_{k1,l}^\theta$ and $d_{k2,l}^\theta$. Table 4 shows that for $d_{k1,l}^\theta$ the RMSE value is 0.0524, MAE is 0.0537, and MAPE is 5.0011% when applying (74). At the same time, the RMSE value is 0.0765, MAE is 0.0816, and MAPE is 7.6516% when applying the expressions from [28], $d_{k1,l}^\theta(i+1) = d_{k1,l}^\theta(i) - \eta \cdot \frac{\partial E_k}{\partial d_{k1,l}^\theta}$, $d_{k2,l}^\theta(i+1) = d_{k2,l}^\theta(i) - \eta \cdot \frac{\partial E_k}{\partial d_{k2,l}^\theta}$. Similarly, for

$d_{k2,l}^\theta$ RMSE is 0.0521, MAE is 0.0536, and MAPE is 5.0008%, and when using expressions from [28], RMSE is 0.0761, MAE is 0.0815, and MAPE is 7.6512%. Thus, the results obtained indicate a decrease in the RMSE, MAE, and MAPE values for the bell-shaped membership function parameters $d_{k1,l}^\theta$ and $d_{k2,l}^\theta$ by 1.46, 1.52, and 1.53, respectively, when applying (74), adaptive speed training, compared with [28].

Table 5. Values of statistical criteria RMSE, MAE, and MAPE.

Parameters	Statistical Criteria Values					
	RMSE		MAE		MAPE	
	Application (74)	Application [28]	Application (74)	Application [28]	Application (74)	Application [28]
$d_{k1,l}^\theta$	0.0524	0.0765	0.0537	0.0816	5.0011	7.6516
$d_{k2,l}^\theta$	0.0521	0.0761	0.0536	0.0815	5.0008	7.6512

Thus, the work experimentally confirmed the feasibility of an adaptive training rate for introducing the NFN AFNN.

The obtained results of bell-shaped function parameters $d_{k1,l}^\theta$ and $d_{k2,l}^\theta$ made it possible to graphically represent the bell-shaped membership function appearance. Figure 9a shows the bell-shaped membership function appearance before training the AFNN NFN using the initial parameters $d_{k1,l}^\theta$ and $d_{k2,l}^\theta$, and Figure 9b shows the same after training the NFN AFNN with refined parameters $d_{k1,l}^\theta$ and $d_{k2,l}^\theta$. The refinement of parameters $d_{k1,l}^\theta$ and $d_{k2,l}^\theta$ of the bell-shaped membership function is justified by “fine-tuning” the fuzzy model of the research object, an NFA modified closed onboard HTE ACS. After parameter optimization, the bell-shaped membership function underwent changes, resulting in a more accurate fit to system data. These changes improved control quality by adapting the function to the controlled object (HTE) specifics.

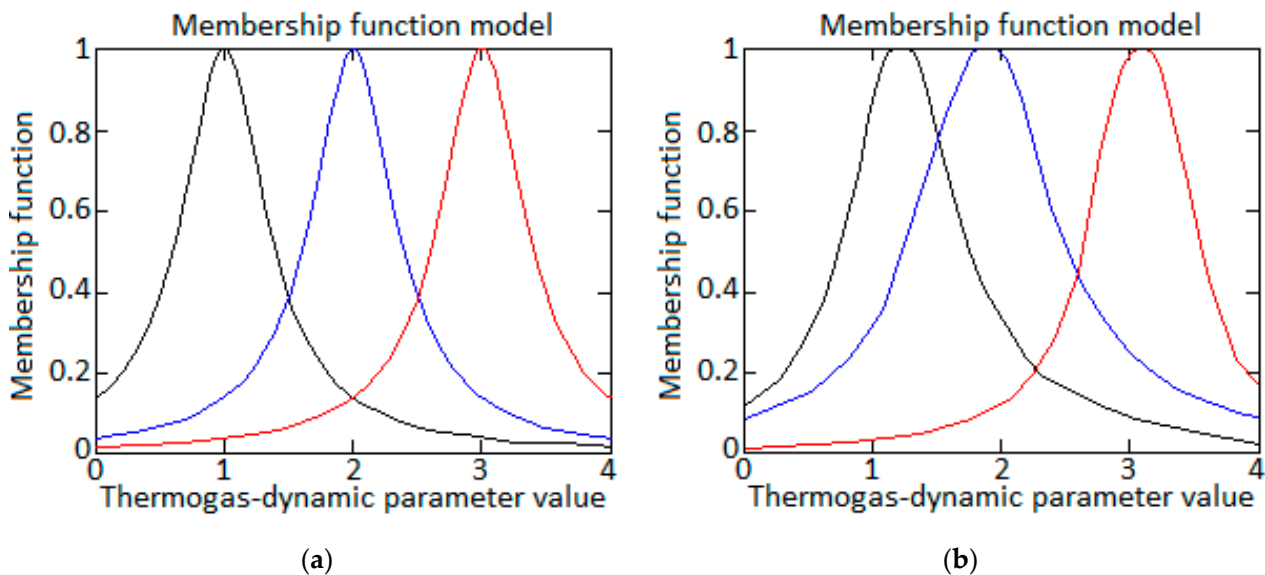


Figure 9. Bell-shaped membership function: (a) before training the NFN AFNN, (b) after training the NFN AFNN [28,52].

3.3. Computational Experiment Results

During the next phase of the computational test, the neuro-fuzzy HTE ACS parameters, which are based on a six-layer neuro-fuzzy AFNN network, were directly determined. As a result of modeling changes in the HTE thermogas-dynamic characteristics according to the training sample data (Table 1) at various moments of model time (from 0 to 5 s), the conclusions are shown in Figure 10, where (a) means the change in n_{TC} , (b) means

the change in T_G^* , and (c) means n_{FT} . The orange curve represents experimental data on TDP recorded on board a helicopter, while the blue curve shows the model values of these parameters after correction using a reconfigured helicopter TE ACS [28], and the black curve shows the model values of these parameters after correction using the proposed neuro-fuzzy HTE ACS (Figure 2).

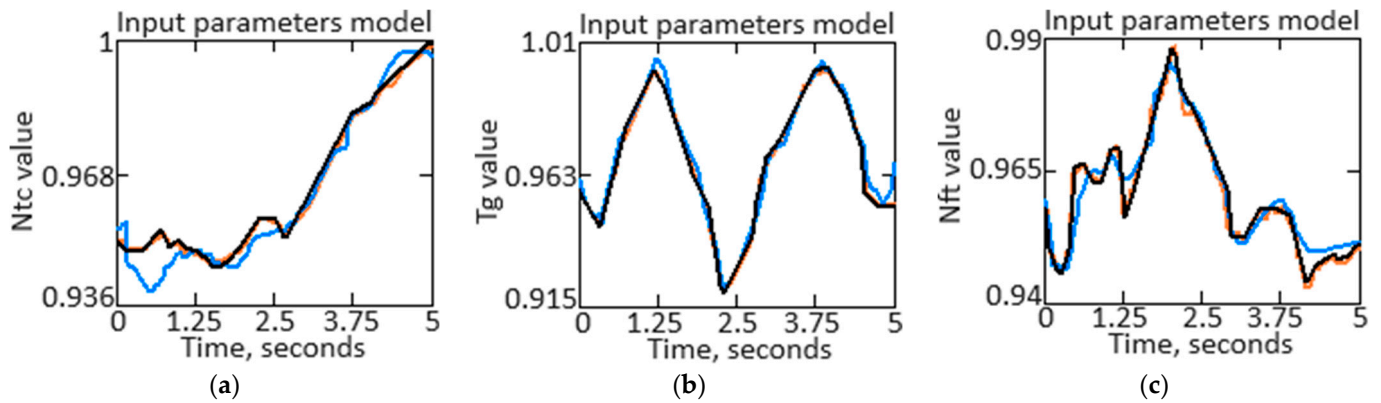


Figure 10. Results of modeling changes in the TE thermogas-dynamic characteristics in the time interval from 0 to 5 s: (a) parameter n_{TC} change, (b) parameter T_G^* change, (c) parameter n_{FT} change.

To confirm the obtained curves' adequacy as a result of modeling, the study used the determination coefficient R^2 as a statistical criterion for comparing the curves. This criterion evaluates the degree of agreement between model (blue curve and black curve) and experimental data (orange curve); a value close to 1 indicates a good fit of model data.

The percentage deviation from the experimental curve can be calculated as the ratio of the difference between the model curve values and the experimental curve to the experimental curve value, multiplied by 100%, according to the expression:

$$R^2 = \sum_{i=1}^n \left| \frac{x_i - \hat{x}_i}{\hat{x}_i} \right| \cdot 100\%, \quad (78)$$

where x_i is the i -th point on the model curves (blue curve and black curve), and \hat{x}_i is the i -th point on the experimental curve (orange curve).

To calculate the determination coefficient R^2 , 50 points were taken on the model and experimental curves with a uniform step of 0.1 s. The determination coefficient for the blue curve obtained using a reconfigured HTE ACS [28] was 0.916, and for the black curve obtained using the proposed neuro-fuzzy HTE ACS (Figure 2)—0.982. Thus, the value of the determination coefficient for the proposed neuro-fuzzy HTE ACS (Figure 2) is 1.07 times higher than the value of the determination coefficient for the reconfigured HTE ACS [28], which indicates better adequacy and accuracy of the proposed HTE ACS.

The change in residuals after regulation $\varepsilon(t) = x_1(t) + x_2(t) + x_3(t) - y(t)$ remains within the HTE TDP permissible deviation, not exceeding 0.004. These parameter dynamics, shown in Figure 11 after regulating $\varepsilon(t)$, demonstrate the control stability, where the $\varepsilon(t)$ value tends to zero.

At the computational experiment's next stage, the fuel consumption parameter G_T was determined (in absolute units) for various control types (Figure 12, where the blue curve shows the reference value of fuel consumption G_T (step action) and the orange curve shows the fuel consumption G_T real value), while, similarly to [28], it was assumed that the fuel consumption parameter G_T reference signal has a rectangular pulse form, that is, a step change in the required fuel consumption G_T .

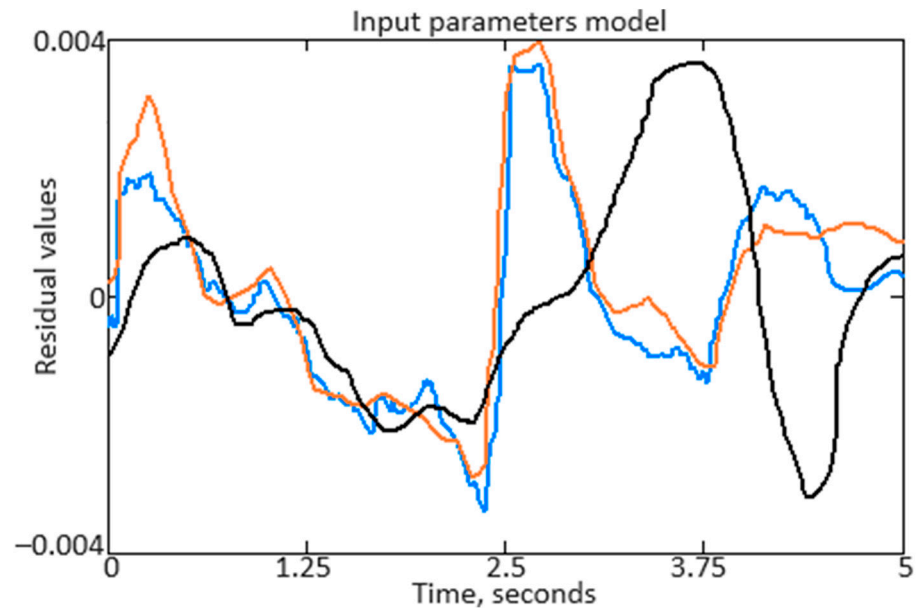


Figure 11. Results of calculating residuals after regulation: orange curve—according to the n_{TC} parameter, blue curve—according to the T_G^* parameter, black curve—according to the n_{FT} parameter [28].

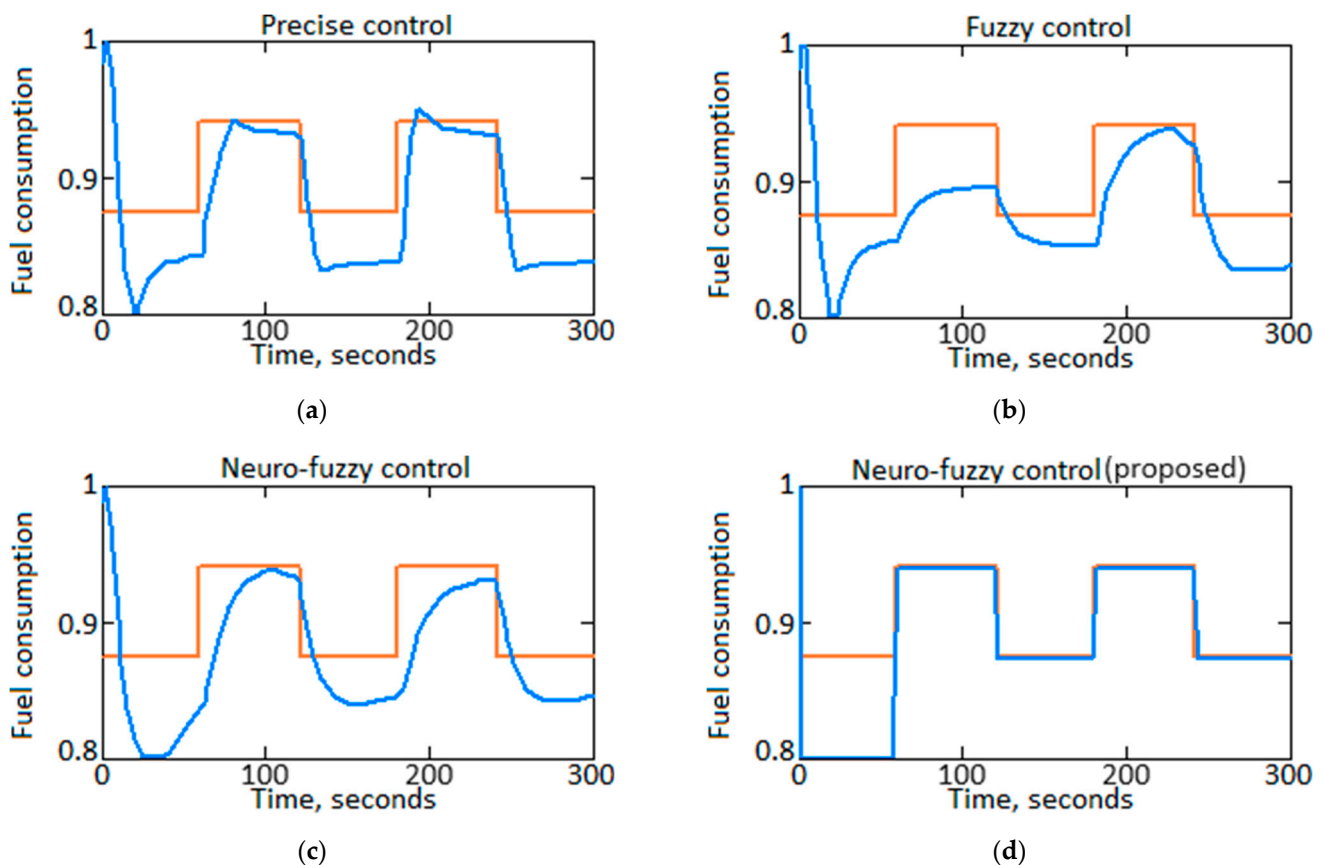


Figure 12. The fuel consumption parameter G_T (in absolute units) calculation results for the HTE TDP given values: (a) for precise control (traditional ACS), (b) for fuzzy control, (c) for neuro-fuzzy control using a reconfigured ACS [28], (d) for neuro-fuzzy control using the proposed helicopter turboshaft engine neuro-fuzzy ACS.

From the presented transient control process analysis diagram (Figure 12), it is clear that the control quality (measured by the duration of the transient process and the maximum

deviation of the controlled value) when using traditional (Figure 12a), fuzzy (Figure 12b), and neuro-fuzzy (Figure 12c) control types are approximately the same. At the same time, when using neuro-fuzzy control using the proposed neuro-fuzzy HTE ACS (Figure 12d), the transient process duration and the controlled value G_T maximum deviation are reduced by an average of 15...20% compared with the above-presented control types. The role of a compensator in neuro-fuzzy control lies in its ability to effectively compensate for uncertainties and disturbances in the ACS. The compensator predicts and corrects the influence of external factors, which allows the system to respond faster and more accurately to changing conditions, which ultimately led to an improvement in the control quality, with a reduction in the duration of the transient process and a reduction in the controlled value G_T by 15...20% maximum deviation.

At the computational experiment's next stage, detailed research on the fuel consumption G_T parameter transition process section is carried out—from 20 to 100 s (Figure 12), where the orange curve is the fuel consumption G_T real value, the black curve shows the transition process for precise control (traditional ACS), the blue curve shows the transition process for fuzzy control, the red curve shows the transition process for neuro-fuzzy control using a reconfigured ACS [28], and the green curve shows the transition process for neuro-fuzzy control using the proposed neuro-fuzzy HTE ACS.

To achieve the ACS stable operation, as well as to fulfill the transient processes' specified quality indicators, according to [28,53–55], the synthesized system parameters' quantities under consideration of dependence are studied:

$$\begin{cases} q_1(\{k_1\}, \{\tau_1\}, \{T_1\}) = f_1(\{k_1\}, \{\tau_1\}, \{T_1\}), \\ q_2(\{k_2\}, \{\tau_2\}, \{T_2\}) = f_2(\{k_2\}, \{\tau_2\}, \{T_2\}), \\ \dots \\ q_i(\{k_i\}, \{\tau_i\}, \{T_i\}) = f_i(\{k_i\}, \{\tau_i\}, \{T_i\}), \end{cases} \quad (79)$$

where q_1, \dots, q_i are the quality indicators of the transient processes under consideration; $\{k_i\}, \{\tau_i\}, \{T_i\}$ are the variable system parameter (gain factors, time constants, etc.) sets; $f_1(\bullet), \dots, f_i(\bullet)$ are the functions expressing the system quality indicators' dependence on the synthesized controllers parameters.

If we take overshoot and subsystem regulation time as the main indicators of system quality, then the system of Equation (79) takes the form:

$$\begin{cases} t_{reg1} = f_1(\{k_1\}, \{\tau_1\}, \{T_1\}), \\ \sigma_1 = g_1(\{k_1\}, \{\tau_1\}, \{T_1\}) \\ \dots \\ t_{reg-i} = f_i(\{k_i\}, \{\tau_i\}, \{T_i\}), \\ \sigma_{reg-i} = g_i(\{k_i\}, \{\tau_i\}, \{T_i\}), \end{cases} \quad (80)$$

where $\sigma_1, \dots, \sigma_i$ is the overshoot, and $t_{reg1}, \dots, t_{reg-i}$ is the subsystems transient processes regulation time.

To determine the subsystems transient processes regulation time according to the transient process diagram (Figure 13), at the initial stage the establishment moment is determined, which indicates the time moment when the output quantity (controlled quantity) approximately stabilizes in the vicinity of the steady-state value. Next is the first point on the diagram (t_1), at which the controlled value enters the vicinity of the steady value and remains in this vicinity further. Next is the last point on the diagram (t_2), at which the controlled value leaves the steady value's vicinity and no longer returns to this vicinity. The regulation time t_{reg} is defined as the difference between t_2 and t_1 , that is:

$$t_{reg} = t_2 - t_1. \quad (81)$$

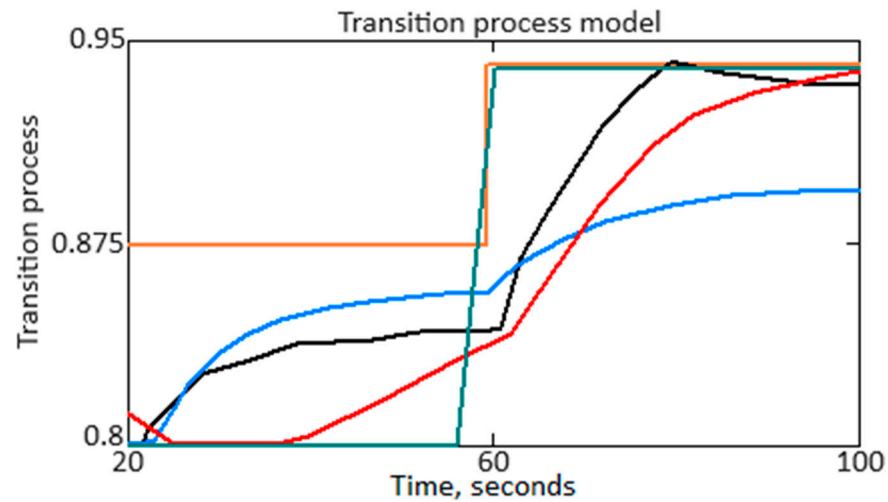


Figure 13. Diagram of the fuel consumption parameter G_T transition process (in absolute units) for the HTE TDP with given values [28].

Table 6 presents the transient process control time calculation results. As can be seen from Table 6, use of the proposed neuro-fuzzy helicopter TE ACS (Figure 2) made it possible to reduce the fuel consumption parameter G_T transient process time by 15.0 times compared with the traditional ACS (precise control) use, by 17.5 times compared with the fuzzy ACS (fuzzy control) use, and 11.25 times compared with the neuro-fuzzy reconfigured ACS [28] use.

Table 6. The transient process control time calculation results.

Precise Control			Fuzzy Control			Neuro-Fuzzy Reconfigured ACS [28]			The Proposed Neuro-Fuzzy ACS (Figure 2)		
t_1	t_2	t_{reg}	t_1	t_2	t_{reg}	t_1	t_2	t_{reg}	t_1	t_2	t_{reg}
20	80	60	23	93	70	37	82	45	56	60	4

The role of the compensator, controller, and emulator in reducing transient control time is to effectively control the system while compensating for disturbances, adapting to changing conditions, and accurately emulating the desired behavior. The compensator in the proposed neuro-fuzzy HTE ACS (Figure 2) allows us to quickly adjust actions to minimize the external factors’ impact, and the controller is able to adapt to changes, providing stable and accurate control. The emulator provides modeling of the system-required behavior, and together these can significantly reduce the time to achieve a transient process steady state compared to traditional control methods.

At the computational experiment’s next stage, using the transition process diagram (Figure 13), the following quality indicators for ACS were determined:

1. Overshoot (overload) is the controlled value (fuel consumption parameter G_T) maximum excess relative to its steady value during the transition to a new state, determined according to the expression:

$$O = \frac{G_{Tmax} - G_{Tst}}{G_{Tst}} \cdot 100\%, \tag{82}$$

where G_{Tmax} is the controlled variable (fuel consumption parameter G_T) maximum value during the transition process, and G_{Tst} is the steady-state value (target value).

2. The steady-state value (stationary deviation) is defined as the difference between the controlled variable (the target value of the fuel consumption parameter G_T) steady-state value and its value after the transient process completion. This indicator allows

us to evaluate the system deviation degree from the desired state after stabilization and is calculated according to the expression:

$$\epsilon_{st} = G_{Tst} - G_T(t), \quad (83)$$

where $G_T(t)$ is the controlled variable (fuel consumption parameter G_T) value at some point in time t after the transition process completion.

3. The transient process time is defined as the time during which the controlled value (fuel consumption parameter G_T) reaches a certain percentage deviation from the steady-state value after a regime violation. The time of the transient process is measured from the moment the control action begins to change until the moment when the controlled value (system output) approaches the steady-state value, and is calculated as:

$$t_{trans} = t_2 - t_1, \quad (84)$$

where t_1 is the point in time when the controlled variable (fuel consumption parameter G_T) first reaches or enters a given vicinity (p , %) of the steady-state value G_{Tst} , and t_2 is the point in time when the controlled variable (fuel consumption parameter G_T) returns to the same neighborhood and is close to the steady-state value after the transition.

Table 7 presents the calculated values for overregulation (overload), steady-state value (stationary deviation), and the transition process time for precise control, fuzzy control, neuro-fuzzy control using a reconfigured ACS [28], and neuro-fuzzy control using the proposed neuro-fuzzy HTE ACS.

Table 7. Control quality indicator calculations results.

Parameter	Precise Control	Fuzzy Control	Neuro-Fuzzy Reconfigured ACS [28]	The Proposed Neuro-fuzzy ACS (Figure 2)
Overshoot (overload)	5%	4.5%	2.8%	1.3%
Steady value	0.0103	0.0076	0.0045	0.0031
Transition time	7.85	6.34	3.18	1.10

Table 7 shows that the use of the proposed neuro-fuzzy helicopter TE ACS (Figure 2) made it possible to achieve an improvement in the fuel consumption parameter G_T control quality compared with clear control, fuzzy control, and neuro-fuzzy control using a reconfigured ACS [28], in terms of overshoot indicators (overload), steady-state value (stationary deviation), and transient process time by 1.45...7.77 times. At the same time, the compensator in the proposed neuro-fuzzy HTE ACS (Figure 2) plays a role in taking into account and compensating for disturbances and uncertainties, which helps reduce overshoot and steady-state deviation. The controller provides adaptive control, optimizing parameters in real time and ensuring fast and accurate response to changes. The emulator simulates the system desired behavior, which helps improve control efficiency and achieve better-quality indicators compared to more traditional control methods. These elements together allow the proposed neuro-fuzzy HTE ACS (Figure 2) to achieve significant improvements in overshoot, steady-state deviation, and transient process time compared to other control types.

At the computational experiment's final stage, the proposed neuro-fuzzy HTE ACS stability is determined (Figure 2) using the transition process diagram (Figure 12, red curve). The use of a stability criterion, such as the Lyapunov method, allows one to assess the dynamic system stability and identify the conditions under which the system remains stable over time. To use the Lyapunov method, according to the transition process (Figure 13) diagram, the Lyapunov function $V(x)$ is determined, where x is the system state vector, which must be positive definite and continuous. Next, the derivative of the Lyapunov

function with respect to time $\frac{\partial V(x)}{\partial t}$ for the system is calculated using the system dynamics equation. That is, for a system with a differential equation $\frac{\partial x}{\partial t} = f(x)$,

$$\frac{\partial V(x)}{\partial t} = \frac{\partial V}{\partial t} = f(x) \cdot \frac{\partial V}{\partial x}. \quad (85)$$

The system is considered asymptotically stable if derivative (85) is negative definite (or not positive definite) in some region x .

According to Figure 13, the approximated curve corresponding to the fuel consumption parameter G_T transient process for the proposed neuro-fuzzy HTE ACS (Figure 2) has the form:

$$G_T(t) = -0.0467 \cdot t^3 + 0.35 \cdot t^2 - 0.7233 \cdot t + 1.22. \quad (86)$$

It is known that the Lyapunov function $V(t)$ must be positive definite. Therefore, the work proposes the use of the Lyapunov function $V(t) = t^2$, which is positive for all $t \neq 0$. To calculate the time derivative of the Lyapunov function $\frac{\partial V(x)}{\partial t}$, we use the system dynamic equation $\frac{\partial x}{\partial t} = f(x)$ (86). The derivative of the Lyapunov function with respect to time is calculated as:

$$\frac{\partial V(x)}{\partial t} = \frac{\partial V}{\partial x} \cdot \frac{\partial x}{\partial t}. \quad (87)$$

For the chosen Lyapunov function, $V(t) = t^2$ $\frac{\partial V}{\partial x} = 2 \cdot t$. Then

$$\frac{\partial V(x)}{\partial t} = 2 \cdot t \cdot f(x) = 2 \cdot t \cdot (-0.0467 \cdot t^3 + 0.35 \cdot t^2 - 0.7233 \cdot t + 1.22) = -0.0934 \cdot t^4 + 0.7 \cdot t^3 - 1.4466 \cdot t^2 + 2.44 \cdot t. \quad (88)$$

The system will be asymptotically stable if $\frac{\partial V(x)}{\partial t}$ is negative definite (or not positive definite). To do this, we investigated the behavior of this expression for different t values:

1. At $t = 0$ $\frac{\partial V(x)}{\partial t} = 0$.
2. For large positive t (for example, $t > 1.22$): the leading term $-0.0934 \cdot t^4$ dominates and $\frac{\partial V(x)}{\partial t} < 0$.
3. At $t < 0$, the leading term $-0.0934 \cdot t^4$ also dominates and $\frac{\partial V(x)}{\partial t} < 0$.

Therefore, when analyzing the Lyapunov function derivative, we see that for all t values (except $t = 0$), the polynomial dominates the leading term, which makes $\frac{\partial V(x)}{\partial t}$ negative definite. Thus, we can conclude that the system is asymptotically stable.

Based on the proposed neuro-fuzzy HTE (Figure 2) stability analysis results, using the Lyapunov method, it was established that the system is asymptotically stable. The derivative of the Lyapunov function with respect to time $\frac{\partial V(x)}{\partial t}$ for a given system is negative definite, with the exception of the moment of time $t = 0$. At the moment $t = 0$, the Lyapunov function derivative with respect to time $\frac{\partial V(x)}{\partial t} = 0$, which means that the system is in an equilibrium state. In this state, if the system is not subject to external disturbances or the initial conditions do not throw it out of equilibrium, then it will remain in this state. It is important to note that $\frac{\partial V(x)}{\partial t} = 0$ indicates a point where the system does not change its state, which is a sign of stability. However, to confirm asymptotic stability, it is necessary that, near this state, $\frac{\partial V(x)}{\partial t}$ be negatively defined, which is confirmed in this case for all values of t except $t = 0$. Thus, at the slightest deviation from the equilibrium point, the system will return to it, which ensures its stability over time.

4. Discussion

The work carried out a proposed neuro-fuzzy HTE ACS (Figure 2) comparative analysis, the basis of which is a six-layer NFN AFNN, with its closest analogue being the reconfigured neuro-fuzzy HTE ACS, the basis of which is a five-layer neuro-fuzzy ANFIS network [28]. Table 8 shows the main advantages of the proposed neuro-fuzzy HTE ACS (Figure 2) over the reconfigured neuro-fuzzy HTE ACS [28] according to the following

quality metrics: NFN type, NFN training algorithm, fuzzy inference type, accuracy, loss, precision, recall, F-score, efficiency coefficient, quality coefficient, correlation coefficient, determination coefficient [38], as well as training speed, resource efficiency use by an NFN, and and robustness.

Table 8. Comparative analysis results of NFN (AFNN or ANFIS) use in the neuro-fuzzy HTE ACS (author’s research).

Index	Six-Layer NFN ANFF (NN_1)	Five-Layer NFN ANFIS (NN_2) [28]	Advantages of NN_1 over NN_2
NFN type	ANFF	ANFIS	The advantage of the six-layer AFNN network over the five-layer ANFIS network is that the model is more flexible due to additional layers, which allows it to more effectively capture and use fuzzy information to control the system.
NFN training algorithm	Modified interactive adaptation method	Traditional training method	The advantage of the modified interactive adaptation method in training a six-layer AFNN network over the classical method of training a five-layer ANFIS network is the increased efficiency of adaptation to changing conditions and the fuzzy system thanks to more accurate control of an additional layer and interactive parameters optimization.
Fuzzy inference type	Sugeno Model	TSK Model	The advantage of Sugeno-type fuzzy inference in a six-layer AFNN network over TSK-type fuzzy inference in a five-layer ANFIS network is the ability to more flexibly and accurately model nonlinear dependencies between input and output data due to a more complex network structure and the inference method specifics.
Accuracy	0.9998 (99.98%)	0.9752 (97.52%)	The Accuracy metric for a six-layer AFNN network is 1.03 times higher than for a five-layer ANFIS network. This means that the six-layer AFNN network has higher accuracy and the ability to correctly classify or predict data compared to the five-layer ANFIS network in the neuro-fuzzy HTE ACS.
Loss	0.0002 (0.2%)	0.0248 (2.48%)	The Loss metric for a six-layer AFNN network is 12.4 times less than for a five-layer ANFIS network. This means that the six-layer AFNN network demonstrates significantly lower losses during the training process compared to the five-layer ANFIS network in a neuro-fuzzy ACS, which indicates its effectiveness and accuracy in modeling TE parameters.
Precision	0.9861 (98.61%)	0.9622 (96.22%)	The Precision metric for the six-layer AFNN network is 1.03 times higher than for the five-layer ANFIS network. This means that the six-layer AFNN network has a slight improvement in accuracy compared to the five-layer ANFIS network in the neuro-fuzzy HTE ACS, which may indicate a slight advantage in correctly predicting positive classes or relevant results.
Recall	1.0	0.988	The Recall metric for the six-layer AFNN network is 1.012 times higher than for the five-layer ANFIS network. This means that the six-layer AFNN network shows a slight improvement in Recall performance compared to the five-layer ANFIS network in the neuro-fuzzy helicopter ACS TE, which indicates a small increase in the model’s ability to detect true positives among all positives, which may be important for tasks, HTE fuel consumption parameter control, and regulation, since it requires high sensitivity to detection.

Table 8. Cont.

Index	Six-Layer NFN ANFF (NN_1)	Five-Layer NFN ANFIS (NN_2) [28]	Advantages of NN_1 over NN_2
F-score	0.993	0.975	<p>The F-score metric for the six-layer AFNN network is 1.019 times higher than for the five-layer ANFIS network. This indicates a slight improvement in the balanced F-score of the six-layer AFNN compared to the five-layer ANFIS network, highlighting the higher balance between AFNN model Precision and Recall.</p> <p>The efficiency coefficient of the six-layer AFNN network is 1.74% higher than that of the five-layer ANFIS network. This means a slight but noticeable improvement in the performance or accuracy of the AFNN model compared to ANFIS, which is important when choosing the optimal model for the HTE fuel consumption parameter controlling and regulating task.</p> <p>The quality coefficient of the six-layer AFNN network is 1.92% higher than that of the five-layer ANFIS network. This means a slight but noticeable improvement in the overall efficiency, accuracy or other quality indicators of the AFNN model compared to ANFIS, which may be important when choosing the best model for controlling the HTE fuel consumption parameter and regulating flight conditions.</p> <p>The correlation coefficient of the six-layer AFNN network is 1.12 times larger than that of the five-layer ANFIS network. This means stronger and more consistent linear relations between the AFNN model predicted and true output values compared to the ANFIS model, highlighting the six-layer network's predictions' higher accuracy or reliability.</p> <p>The determination coefficient of the six-layer AFNN network is 1.12 times larger than that of the five-layer ANFIS network. This means the AFNN model has a greater ability to explain the target variable variation compared to the ANFIS model, indicating a more efficient use of input data to predict outputs and a more accurate modeling of the HTE fuel consumption parameter dependence on TDPs.</p> <p>It should be noted that the difference in training time between the six-layer AFNN network and the five-layer ANFIS network is only 0.035 s, which is insignificant and has virtually no effect on the neuro-fuzzy HTE ACS model's overall effectiveness or applicability.</p> <p>The efficiency metric for using computing resources for a six-layer AFNN network is 1.02 times higher than for a five-layer ANFIS network. This represents a small but noticeable improvement in the computational efficiency of the AFNN model compared to ANFIS. This may indicate a more optimal use of computing power or faster convergence of the AFNN model during training, which is important for use in resource-dependent scenarios or on devices with limited computing capabilities, for example, an onboard neural network expert system for complex monitoring of HTE, implemented on an Intel processor Neural Compute Stick 2.</p>
Efficiency coefficient	0.9902 (99.02%)	0.9728 (97.28%)	
Quality coefficient	0.9901 (99.01%)	0.9709 (97.09%)	
Correlation coefficient	0.9923	0.8841	
Determination coefficient	0.9945	0.8897	
Training rate	4.061 s	4.026 s	
Resource efficiency	0.2462	0.2422	

Table 8. Cont.

Index	Six-Layer NFN ANFF (NN_1)	Five-Layer NFN ANFIS (NN_2) [28]	Advantages of NN_1 over NN_2
Robustness	0	$7.071 \cdot 10^{-5}$	The accuracy standard deviation, which determines the robustness of NFNs as part of the neuro-fuzzy HTE ACS, for a six-layer AFNN network is equal to zero, while the robustness of a five-layer ANFIS network is different from zero. This means that the accuracy of the AFNN model is stable and does not change under different conditions, while the accuracy of the ANFIS model can change and be more susceptible to variations in the input data of the HTE TDP under flight operating conditions.
Response time	12.86 ms	25.19 ms	The response time of the six-layer AFNN network is 1.96 times larger than that of the five-layer ANFIS network, indicating a potential trade-off between the increased model complexity and the computational efficiency in real-time applications.
Rate of error reduction	0.00132 s^{-1}	0.00062 s^{-1}	The rate of error reduction of the six-layer AFNN network is 2.13 times larger than that of the five-layer ANFIS network, suggesting that the additional layer in the AFNN architecture enhances its adaptive capability, allowing it to correct errors more efficiently in real-time scenarios.
Conclusion	The AFNN six-layer NFN demonstrates higher model flexibility and efficiency when using fuzzy information to control the fuel consumption parameter compared to the ANFIS five-layer NFN in the neuro-fuzzy HTE ACS, which emphasizes its advantage in the context of accuracy and flight operation adaptability to helicopter conditions, especially in a changing and uncertain environment.		

The accuracy of the controlled variable (fuel consumption parameter G_T) values determined by the test data set is a measure of the NFN (AFNN or ANFIS) output values' correspondence to the expected G_T values based on the provided test data [56–58]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (89)$$

where TP (True Positive) is the true positive results number, TN (True Negative) is the true negative results number, FP (False Positive) is the false positive results number, and FN (False Negative) is the false negative results number (Table 9).

Table 9. Error matrix [49].

Classification	Dataset Class Labels	
	Positive Grade Label	Negative Grade Label
Class labels exposed by the NFN	Positive grade label	TP
	Negative grade label	FN

Precision is a classification quality metric that measures the proportion of control variable (fuel consumption G_T parameter) values that actually belong to the positive class among all objects that were predicted to be positive. The higher the precision, the fewer false positives the model produces, which means that the model is less likely to misclassify objects of a negative class as positive. High precision is critical, especially when the FP error (false positive) is high-cost. Precision is defined as:

$$Precision = \frac{TP}{TP + FP} \quad (90)$$

Recall is a metric used to evaluate the classification quality, measuring the proportion of positive class objects that the NFN (AFNN or ANFIS) correctly identified as positive. Recall is defined as:

$$Recall = \frac{TP}{TP + FN}. \quad (91)$$

F-score is a metric used to evaluate test accuracy, which is the harmonic mean between Precision and Recall. F-score is defined as:

$$F - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}. \quad (92)$$

The efficiency coefficient evaluates the NFN (AFNN or ANFIS) training efficiency and is defined as the ratio of the change in the loss function at the current iteration to the change in the network parameters at the same iteration:

$$K_{eff} = \frac{|E(\theta_k) - E(\theta_{k-1})|}{\|\theta_k - \theta_{k-1}\|}, \quad (93)$$

where $E(\theta_k)$ is the loss function value at the current iteration, $E(\theta_{k-1})$ is the loss function value at the previous iteration, and $\|\theta_k - \theta_{k-1}\|$ is the rate of change in the NFN (AFNN or ANFIS) parameters at the current iteration.

The quality factor evaluates the NFN (AFNN or ANFIS) parameters' approximation accuracy and is defined as the ratio of the reduction in the loss function at the current iteration to the total loss function at previous iterations:

$$K_{quality} = \frac{E(\theta_{k-1}) - E(\theta_k)}{E(\theta_0) - E(\theta_{k-1})}, \quad (94)$$

where $E(\theta_0)$ is the loss function initial value.

The correlation coefficient is a statistical measure used to assess the degree of linear relations between actual and calculated using NFN (AFNN or ANFIS) values of the controlled variable (fuel consumption parameter G_T):

$$R = 1 - \frac{\sum_{i=1}^n (G_{T_i} - \bar{G}_T) \cdot (\hat{G}_{T_i} - \bar{\hat{G}}_T)}{\sum_{i=1}^n (G_{T_i} - \bar{G}_T) \cdot \sum_{i=1}^n (\hat{G}_{T_i} - \bar{\hat{G}}_T)}, \quad (95)$$

where G_{T_i} is the controlled variable (fuel consumption parameter G_T) actual value, \hat{G}_{T_i} is the controlled variable (fuel consumption parameter G_T) approximated value for the i -th example, \bar{G}_T is the controlled variable (parameter fuel consumption G_T) actual values' average value, and $\bar{\hat{G}}_T$ is the controlled variable (fuel consumption parameter G_T) approximated values' average value.

The determination coefficient is a statistical measure that estimates the control variable (fuel consumption parameter G_T) variance proportion explained or accounted for by a NFN (AFNN or ANFIS) and indicates how well the independent variables n_{TC} , T_G^* , and n_{FT} model G_T . The determination coefficient is calculated as:

$$R^2 = 1 - \frac{\sum_{i=1}^n (G_{T_i} - \hat{G}_{T_i})^2}{\sum_{i=1}^n (G_{T_i} - \bar{G}_T)^2}. \quad (96)$$

To assess the resource efficiency of NFN (AFNN or ANFIS) use, the work uses the Efficiency metric, which takes into account the prediction accuracy and the amount of resources spent on training and/or prediction. One such metric is the relationship between accuracy and resource utilization. Taking into account that Acc_1 and Acc_2 are the accuracies of the two compared NFNs (AFNN and ANFIS), and that Res_1 and Res_2 are the resources

used (in this work, the network training time parameter is taken with the same memory amount of 32 GB DDR-4), resource efficiency is defined as:

$$Eff_1 = \frac{Acc_1}{Res_1}, Eff_2 = \frac{Acc_2}{Res_2}. \tag{97}$$

Assessing NFN (AFNN or ANFIS) robustness can be achieved by analyzing its performance on different datasets or under different conditions. One of the approaches to determining robustness in relation to the task being solved is to research the change in the accuracy of the controlled variable (fuel consumption parameter G_T) values when varying the input data or parameters. This study uses the traditional method of assessing robustness, which is based on determining the standard deviation of the determination accuracy values of the controlled variable (fuel consumption parameter G_T) on various subsamples or data sets. Moreover, the smaller the standard deviation, the more resistant the NFN (AFNN or ANFIS) is to changes in data or conditions. The accuracy of controlled variable (fuel consumption parameter G_T) values determining standard deviation is defined as:

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (Acc_i - \overline{Acc})^2}{n}}, \tag{98}$$

where Acc_i is the NFN (AFNN or ANFIS) accuracy on the i -th data set or with the i -th parameter variation, \overline{Acc} is the average accuracy value for all observations, and n is the subsample number.

In this work, for each NFN (AFNN or ANFIS), the accuracy is determined on eight different subsamples of 32 values in each training sample with a total size of 256 values. It is assumed that Acc_{1i} and Acc_{2i} are the accuracy values of the first (AFNN) and second (ANFIS) NFN on the i -th subsample, respectively (Table 10). Thus the average accuracy for each neural network is, respectively, $\overline{Acc}_1 = 0.988$ and $\overline{Acc}_2 = 0.901$, and the average deviation, respectively, is 0.00524 and 0.01148.

Table 10. Calculating accuracy results indicators when dividing a training sample of 256 elements into eight subsamples of 32 elements each (author’s research).

Subsample Number	1	2	3	4	5	6	7	8
NN_1	0.9997	0.9998	0.9999	0.9999	0.9997	0.9998	0.9998	0.9998
NN_2	0.9738	0.9758	0.9742	0.9762	0.9761	0.9754	0.9744	0.9757

Response time is the time interval between the arrival of a new input signal G_{T_i} and the moment when the neuro-fuzzy network updates its output values \hat{G}_{T_i} and adapts its parameters $\theta(t)$ in response to this signal. The response time for the neuro-fuzzy network is calculated as follows:

$$T_{response} = T_{input} + T_{fuzzification} + T_{inference} + T_{defuzzification} + T_{update}, \tag{99}$$

where T_{input} represents the time required for receiving and preprocessing input data; $T_{fuzzification}$ denotes the time needed to perform fuzzification of the input data; $T_{inference}$ indicates the time spent on logical inference based on fuzzy rules; $T_{defuzzification}$ is the time for defuzzification to obtain a crisp output value; and T_{update} refers to the time for network parameter adaptation, including weight and membership function updates.

The error reduction rate $L(t)$ during real-time network parameter adaptation is determined by the error function time derivative and is calculated as:

$$Rate\ of\ Error\ Reduction = -\frac{dL(t)}{dt}, \tag{100}$$

where $L(t)$ is the error function at time t , and $\frac{dL(t)}{dt}$ represents the time derivative of the error function, indicating how quickly the error changes over time. The negative sign before the derivative indicates a decrease in error. A larger magnitude of $-\frac{dL(t)}{dt}$ signifies that the network adapts more quickly and the error decreases more quickly in real time.

Thus, based on the results of a comparative analysis (Table 8), the effectiveness of using the proposed neuro-fuzzy HTE ACS, the basis of which is the six-layer NFN AFNN, has been confirmed, in comparison with the reconfigured HTE ACS, the basis of which is the five-layer NFN ANFIS.

Future research prospects in evaluating the effectiveness of the proposed method in real-time applications lie in enhancing the accuracy of performance metrics and refining adaptation algorithms. This involves investigating advanced techniques for reducing response time, improving error reduction rates, and optimizing computational efficiency. Further studies should also explore the integration of adaptive learning mechanisms to address dynamic conditions and the impact of varying operational environments on system performance. Additionally, examining the scalability of the method for larger and more complex networks could provide insights into its broader applicability and robustness.

Table 11 comparing results for the proposed approach with clear, fuzzy, and neuro-fuzzy control according to the quality metrics used in the article. As can be seen from Table 11, the use of the proposed approach is advisable according to the studied quality metrics (accuracy, loss, precision, recall, F-score) in comparison with precise control, fuzzy control, and neuro-fuzzy reconfigured ACS [28].

Table 11. Comparative analysis results (author's research).

Metric	Precise Control	Fuzzy Control	Neuro-Fuzzy Reconfigured ACS [28]	The Proposed Neuro-Fuzzy ACS (Figure 2)
Accuracy	0.8835 (88.35%)	0.9246 (92.46%)	0.9752 (97.52%)	0.9998 (99.98%)
Loss	0.1165 (11.65%)	0.0754 (7.54%)	0.0248 (2.48%)	0.0002 (0.2%)
Precision	0.8692 (86.92%)	0.9117 (91.17%)	0.9622 (96.22%)	0.9861 (98.61%)
Recall	0.909	0.954	0.988	1.0
F-score	0.889	0.931	0.975	0.993

The proposed neuro-fuzzy HTE ACS in the form of a six-layer NFN AFNN is implemented in software in the Borland Delphi 7 environment and tested using the example of G_T fuel consumption control according to the following input variables: n_{TC} , T_G^* , n_{FT} . The output variable was the fuel consumption value G_T , which was calculated as:

$$G_T = \sqrt{\frac{\sum_{i=1}^N (Acc_i - \overline{Acc})^2}{n}}. \quad (101)$$

Figure 14 shows the AFNN NFN training diagram for testing the G_T fuel consumption control model. Analysis of the simulation results allows us to conclude that the model has a sufficient accuracy degree: at epoch 93, the fuel consumption G_T output value error became less than acceptable.

At the final stage of discussing the obtained results, assessment and analysis of the errors of types I and II, which are important in statistics when making statistical decisions, are carried out. The errors of type I occur when we reject a true null hypothesis. This means that a false assumption is made about the difference between the actual and model values of the controlled variable (fuel consumption parameter G_T), when in fact there is no difference. The error of type I is associated with the significance level α , which defines the critical region for the test and represents the maximum acceptable probability of rejecting a true null hypothesis. The errors I type is calculated as:

$$\alpha = P(\text{reject } H_0 | H_0 \text{ true}). \quad (102)$$

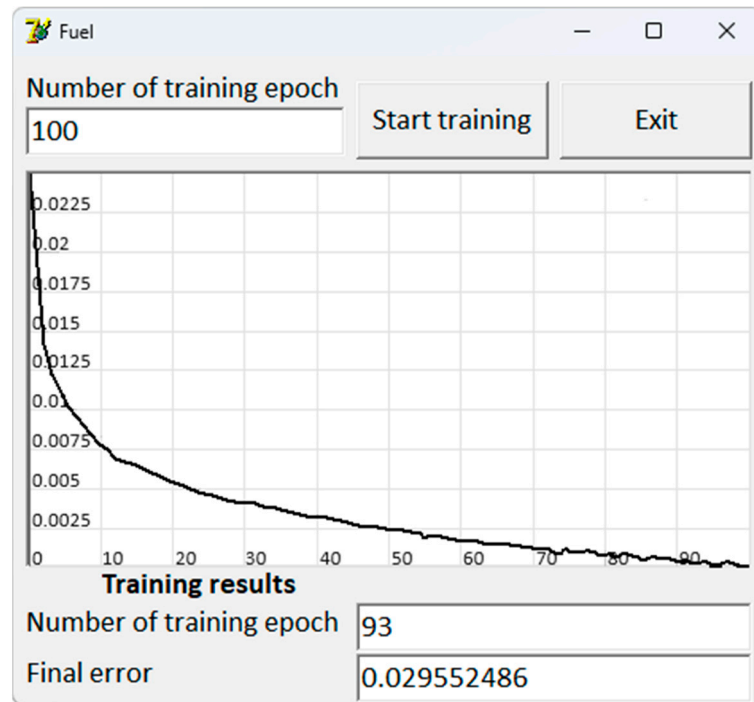


Figure 14. Fuel consumption control model test results: OX axis represents the number of training epochs, and the OY axis represents the training error.

When determining the controlled variable (fuel consumption parameter G_T) values, the test results must show with high confidence a statistically significant difference between the actual and model values of the controlled variable (fuel consumption parameter G_T); the significance level $\alpha = 0.01$ was adopted in this work. This means that if the null hypothesis is true, then the probability of rejecting it (committing the error of type I) is only 1%.

Null hypothesis H_0 : there is no difference between the actual and the fuel consumption parameter G_T model values, that is, the G_T actual value is equal to the G_T model value.

The error of type II occurs when we do not reject a false null hypothesis. This means that we do not detect the presence of an effect or difference when it actually exists. Type II error is related to the power of the test ($1 - \beta$), where the power of the test is the probability of correctly rejecting a false null hypothesis (or detecting an effect if one exists). Equation for the error of type II:

$$\beta = P(\text{do not reject } H_0 | H_0 \text{ false}) = 1 - \beta. \tag{103}$$

where the test “power” concept is defined as the probability of rejecting the null hypothesis when it is in fact false.

Table 12 shows the errors of types I and II in determining the controlled quantity (fuel consumption parameter G_T) values using the proposed neuro-fuzzy HTE ACS, the basis of which is a six-layer NFN AFNN, as well as a reconfigured HTE ACS, the basis of which is a five-layer NFN ANFIS [28].

Table 12. The type I and II calculation errors.

Neuro Fuzzy ACS	Probability of Error in Determining the Optimal Parameters G_T , %	
	The Errors of Type I	The Errors of Type II
Six-layer NFN ANFF	0.17	0.11
Five-layer NFN ANFIS [28]	0.35	0.15

Thus, the use of the ANFF six-layer NFN made it possible to reduce the errors of type I by 2.06 times and the errors of type II by 1.36 times compared with the ANFIS five-layer NFN [28] as part of the neuro-fuzzy HTE ACS.

5. Conclusions

The scientific novelty of the obtained results lies in the development of the helicopter TE modified onboard neuro-fuzzy adaptive ACS, featuring a framework consisting of the research object, regulator, emulator, compensator, and observer block. Each component of this framework utilizes the proposed six-layer hybrid neuro-fuzzy network of the AFNN type, with Sugeno fuzzy inference and bell-shaped membership functions for fuzzy variables.

The NFA closed onboard TE ACS has been improved, which, by introducing an observer unit, taking into account dynamic changes in external conditions, and additional errors (noise), allows us to fulfill the task of controlling the TE fuel consumption.

In contrast to the well-known neuro-fuzzy TE ACS, which is based on an ANFIS five-layer hybrid NFN, it is proposed to use an AFNN six-layer hybrid NFN with Sugeno fuzzy inference and a fuzzy-variables bell-shaped membership function, which made it possible to reduce the HTE fuel consumption parameter transient process regulation time by 15.0 times compared with the use of a traditional ACS (precise control), by 17.5 times compared with the use of a fuzzy ACS (fuzzy control), and by 11.25 times compared with the use of neural fuzzy reconfigured ACS based on an ANFIS five-layer hybrid NFN.

The AFNN six-layer hybrid NFN training algorithm has been further developed, which, due to adaptive elements, allows us to change its parameters and settings in real time based on changing conditions or external influences and, as a result, achieve an accuracy of up to 99.98% in the TE fuel consumption controlling task and reduce losses to 0.2%.

It has been experimentally proven that the use of adaptive elements in the AFNN six-layer hybrid NFN proposed training algorithm has made it possible to reduce the mean square error square root values, the mean absolute error values, and the mean absolute percentage error values in determining the fuzzy-variables bell-shaped membership function parameters by 1.46 . . 1.53 times compared with the use of an ANFIS five-layer hybrid NFN.

It has been experimentally proven that the use of the developed neuro-fuzzy TE ACS has made it possible to achieve an improvement in the fuel consumption parameter control quality in comparison with the precise control, fuzzy control, neuro-fuzzy control using a reconfigured ACS based on an ANFIS five-layer hybrid NFN, in terms of overshoot (overload), steady-state value (stationary deviation), and transition process time by 1.45 . . 7.77 times.

The use of the ANFF six-layer NFN made it possible to reduce the I and II types of error in the TE fuel consumption controlling task by 1.36 . . 2.06 times compared with the ANFIS five-layer NFN.

Author Contributions: Conceptualization, S.V., R.Y. and V.V.; methodology, S.V.; software, V.V.; validation, R.Y., M.B. and V.V.; formal analysis, S.V.; investigation, M.B.; resources, S.V., R.Y. and V.V.; data curation, M.B.; writing—original draft preparation, S.V., R.Y. and V.V.; writing—review and editing, S.V., R.Y. and V.V.; visualization, M.B.; supervision, S.V.; project administration, R.Y.; funding acquisition, M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article.

Acknowledgments: This research was supported by the Ministry of Internal Affairs of Ukraine “Theoretical and applied aspects of the development of the aviation sphere” under Project No. 0123U104884.

Conflicts of Interest: The authors declare no conflicts of interest.

Nomenclature

$d(i)$ is the additional parameter

$y(i)$ is the output signal

$u(i)$ is the control

$f(\bullet)$ is some nonlinear function having known orders r, s, q

$e(i)$ is the control error

X_k is the the fuzzy set

μ is the membership function

θ is the rules number

α is the coefficient adjustable during network training or determined by expert input

J is the relative control error value

J_{nom} is the relative control error nominal value

φ_k is the coefficient that is a fairly large number

λ is the regularization parameter

$g(\bullet)$ is the emulator function

J_{reg} is the emulator average relative error

ϑ is the coefficient that regulates the influence of the adaptive multiplier

η_t is the current adaptive training rate

$v(i)$ is a noise or disturbances vector

$\hat{x}(i|i)$ is the system state current assessment

A and B are the system or control matrices

C is the measurement matrix

$K(i)$ is the observer gain matrix

$P(i)$ is the state covariance error estimate

R is the measurement noise covariance matrix

μ_j are the initial centroids

\hat{y} is the model outputs vector before applying softmax for input x

\hat{y}_k is the vector \hat{y} k -th element

K is the classes number

q_1, \dots, q_i are the transient processes under consideration quality indicators

$\{k_i\}, \{\tau_i\}, \{T_i\}$ are the variable system parameters (gain factors, time constants, etc.) sets

$f_1(\bullet), \dots, f_i(\bullet)$ are the functions expressing the system quality indicators dependence on the synthesized controllers parameters

$\sigma_1, \dots, \sigma_i$ is the overshoot

$t_{reg1}, \dots, t_{reg-i}$ is the subsystems transient processes regulation time

G_T is the fuel consumption parameter

$E(\theta_k)$ is the loss function value at the current iteration

$E(\theta_0)$ is the loss function initial value

Acc_i is the NFN (AFNN or ANFIS) accuracy on the i -th data set or with the i -th parameters variation

Acc is the average accuracy value for all observations

n is the subsamples number.

References

1. Mao, X.; Xiang, Y. An efficient nonlinear adaptive filter algorithm based on the rectified linear unit. *Digit. Signal Process.* **2024**, *146*, 104373. [\[CrossRef\]](#)
2. Jiang, Y.; Liu, L.; Feng, G. Fully distributed adaptive control for output consensus of uncertain discrete-time linear multi-agent systems. *Automatica* **2024**, *162*, 111531. [\[CrossRef\]](#)
3. Liu, J. Gas path fault diagnosis of aircraft engine using HELM and transfer learning. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105149. [\[CrossRef\]](#)
4. Xu, M.; Wang, J.; Liu, J.; Li, M.; Geng, J.; Wu, Y.; Song, Z. An improved hybrid modeling method based on extreme learning machine for gas turbine engine. *Aerosp. Sci. Technol.* **2020**, *107*, 106333. [\[CrossRef\]](#)
5. Montazeri-Gh, M.; Yazdani, S. Application of interval type-2 fuzzy logic systems to gas turbine fault diagnosis. *Appl. Soft Comput.* **2020**, *96*, 106703. [\[CrossRef\]](#)
6. Sarwar, U.; Muhammad, M.; Mokhtar, A.A.; Khan, R.; Behrani, P.; Kaka, S. Hybrid intelligence for enhanced fault detection and diagnosis for industrial gas turbine engine. *Results Eng.* **2024**, *21*, 101841. [\[CrossRef\]](#)
7. Yazdani, S.; Montazeri-Gh, M. A novel gas turbine fault detection and identification strategy based on hybrid dimensionality reduction and uncertain rule-based fuzzy logic. *Comput. Ind.* **2020**, *115*, 103131. [\[CrossRef\]](#)

8. Djeddi, C.; Hafaiifa, A.; Iratni, A.; Hadroug, N.; Chen, X. Robust diagnosis with high protection to gas turbine failures identification based on a fuzzy neuro inference monitoring approach. *J. Manuf. Syst.* **2021**, *59*, 190–213. [[CrossRef](#)]
9. Gu, Z.; Li, Q.; Pang, S.; Zhou, W.; Wu, J.; Zhang, C. Turbo-shaft engine adaptive neural network control based on nonlinear state space equation. *Chin. J. Aeronaut.* **2024**, *37*, 493–507. [[CrossRef](#)]
10. Baranovskyi, D.; Myamlin, S.; Kebal, I. Increasing the Carrying Capacity of the Solid-Body Rail Freight Car. *Adv. Sci. Technol. Res. J.* **2022**, *16*, 219–225. [[CrossRef](#)]
11. Tereikovskiy, I.; Hu, Z.; Chernyshev, D.; Tereikovska, L.; Korystin, O.; Tereikovskiy, O. The Method of Semantic Image Segmentation Using Neural Networks. *Int. J. Image Graph. Signal Process.* **2022**, *14*, 1–14. [[CrossRef](#)]
12. Vladov, S.; Yakovliev, R.; Hubachov, O.; Rud, J.; Stushchanskyi, Y. Neural Network Modeling of Helicopters Turboshift Engines at Flight Modes Using an Approach Based on “Black Box” Models. *CEUR Workshop Proc.* **2024**, *3624*, 116–135.
13. Baranovskyi, D.; Myamlin, S. The criterion of development of processes of the self organization of subsystems of the second level in tribosystems of diesel engine. *Sci. Rep.* **2023**, *13*, 5736. [[CrossRef](#)]
14. Baranovskyi, D.; Bulakh, M.; Michajlyszyn, A.; Myamlin, S.; Muradian, L. Determination of the Risk of Failures of Locomotive Diesel Engines in Maintenance. *Energies* **2023**, *16*, 4995. [[CrossRef](#)]
15. Baranovskyi, D.; Myamlin, S.; Podosonov, D.; Muradian, L. Determination of the filler concentration of the composite material to reduce the wear of the central bowl of the rail truck bolster. *Ain Shams Eng. J.* **2023**, *12*, 102232. [[CrossRef](#)]
16. Baranovskyi, D.; Bulakh, M.; Myamlin, S.; Kebal, I. New Design of the Hatch Cover to Increase the Carrying Capacity of the Gondola Car. *Adv. Sci. Technol. Res. J.* **2022**, *16*, 186–191. [[CrossRef](#)]
17. Rusyn, B.; Lutsyk, O.; Kosarevych, R.; Maksymyuk, T.; Gazda, J. Features extraction from multi-spectral remote sensing images based on multi-threshold binarization. *Sci. Rep.* **2023**, *13*, 19655. [[CrossRef](#)] [[PubMed](#)]
18. Kosarevych, R.; Lutsyk, O.; Rusyn, B. Detection of pixels corrupted by impulse noise using random point patterns, The Visual Computer. *Int. J. Comput. Graph.* **2022**, *38*, 3719–3730. [[CrossRef](#)]
19. Bulakh, M.; Klich, L.; Baranovska, O.; Baida, A.; Myamlin, S. Reducing Traction Energy Consumption with a Decrease in the Weight of an All-Metal Gondola Car. *Energies* **2023**, *16*, 6733. [[CrossRef](#)]
20. Salilew, W.M.; Abdul Karim, Z.A.; Lemma, T.A. Investigation of fault detection and isolation accuracy of different Machine learning techniques with different data processing methods for gas turbine. *Alex. Eng. J.* **2022**, *61*, 12635–12651. [[CrossRef](#)]
21. Hashmi, M.B.; Mansouri, M.; Assadi, M. Dynamic performance and control strategies of micro gas turbines: State-of-the-art review, methods, and technologies. *Energy Convers. Manag. X* **2023**, *18*, 100376. [[CrossRef](#)]
22. Kilic, U.; Villareal-Valderrama, F.; Ayar, M.; Ekici, S.; Amezcua-Brooks, L.; Hikmet Karakoc, T. Deep learning-based forecasting modeling of micro gas turbine performance projection: An experimental approach. *Eng. Appl. Artif. Intell.* **2024**, *130*, 107769. [[CrossRef](#)]
23. Chen, Q.; Sheng, H.; Liu, T. Fuzzy logic-based adaptive tracking weight-tuned direct performance predictive control method of aero-engine. *Aerosp. Sci. Technol.* **2023**, *140*, 108494. [[CrossRef](#)]
24. Baranovskyi, D.; Myamlin, S.; Bulakh, M.; Podosonov, D.; Muradian, L. Determination of the Filler Concentration of the Composite Tape. *Appl. Sci.* **2022**, *12*, 11044. [[CrossRef](#)]
25. Feng, H.; Liu, B.; Xu, M.; Li, M.; Song, Z. Model-based deduction learning control: A novel method for optimizing gas turbine engine afterburner transient. *Energy* **2024**, *292*, 130512. [[CrossRef](#)]
26. Vladov, S.; Shmelov, Y.; Yakovliev, R. Helicopters Aircraft Engines Self-Organizing Neural Network Automatic Control System. *CEUR Workshop Proc.* **2022**, *3137*, 28–47. [[CrossRef](#)]
27. Vladov, S.; Shmelov, Y.; Yakovliev, R.; Petchenko, M. Neural Network Method for Parametric Adaptation Helicopters Turboshift Engines On-Board Automatic Control System Parameters. *CEUR Workshop Proc.* **2023**, *3403*, 179–195.
28. Vladov, S.; Shmelov, Y.; Yakovliev, R.; Kozlovskaya, T.; Petchenko, M. Helicopters Turboshift Engines Intelligent Control Algorithms Synthesis, Taking into Account Required Quality Provision. *CEUR Workshop Proc.* **2023**, *3468*, 71–90.
29. Siddikov, I.X.; Umurzakova, D.M. Synthesis of adaptive neuro-fuzzy control system of nonlinear dynamic objects. *Railw. Transp. Top. Issues Innov.* **2019**, *1–2*, 54–73.
30. Sheng, H.; Chen, Q.; Li, J.; Jiang, W.; Wang, Z.; Liu, Z.; Zhang, T.; Liu, Y. Research on dynamic modeling and performance analysis of helicopter turboshift engine’s start-up process. *Aerosp. Sci. Technol.* **2020**, *106*, 106097. [[CrossRef](#)]
31. Turan, Ö.; Aydın, H. Numerical calculation of energy and exergy flows of a turboshift engine for power generation and helicopter applications. *Energy* **2016**, *115*, 914–923. [[CrossRef](#)]
32. Togni, G.; Nikolaidis, T.; Sampath, D. A combined technique of Kalman filter, artificial neural network and fuzzy logic for gas turbines and signal fault isolation. *Chin. J. Aeronaut.* **2021**, *34*, 124–135. [[CrossRef](#)]
33. Pasięka, M.; Grzesik, N.; Kuźma, K. Simulation modeling of fuzzy logic controller for aircraft engines. *Int. J. Comput.* **2017**, *16*, 27–33. [[CrossRef](#)]
34. Gao, J.; Heng, F.; Yuan, Y.; Liu, Y. A novel machine learning method for multiaxial fatigue life prediction: Improved adaptive neuro-fuzzy inference system. *Int. J. Fatigue* **2024**, *178*, 108007. [[CrossRef](#)]
35. Shihabudheen, K.V.; Pillai, G.N. Recent advances in neuro-fuzzy system: A survey. *Knowl.-Based Syst.* **2018**, *152*, 136–162. [[CrossRef](#)]
36. Gudla, S.; Rao, K.N. Reliable Data Delivery Using Fuzzy Reinforcement Learning in Wireless Sensor Networks. *Int. J. Comput. Netw. Inf. Secur.* **2023**, *15*, 96–107. [[CrossRef](#)]

37. Chang, Q.; Zhang, Z.; Wei, F.; Wang, J.; Pedrycz, W.; Pal, N.R. Adaptive Nonstationary Fuzzy Neural Network. *Knowl.-Based Syst.* **2024**, *288*, 111398. [[CrossRef](#)]
38. Zhang, Y.; Wang, G.; Huang, X.; Ding, W. TSK fuzzy system fusion at sensitivity-ensemble-level for imbalanced data classification. *Inf. Fusion* **2023**, *92*, 350–362. [[CrossRef](#)]
39. Vladov, S.; Yakovliev, R.; Hubachov, O.; Rud, J. Neuro-Fuzzy System for Detection Fuel Consumption of Helicopters Turboshaft Engines. *CEUR Workshop Proc.* **2023**, *3628*, 628–647.
40. Marakhimov, A.R.; Khudaybergenov, K.K. Approach to the synthesis of neural network structure during classification. *Int. J. Comput.* **2020**, *19*, 20–26. [[CrossRef](#)]
41. Hu, Z.; Buriachok, V.; TajDini, M.; Sokolov, V. Authentication System by Human Brainwaves Using Machine Learning and Artificial Intelligence. *Lect. Notes Data Eng. Commun. Technol.* **2021**, *83*, 374–388. [[CrossRef](#)]
42. Liu, J.; Zhao, T.; Cao, J.; Li, P. Interval type-2 fuzzy neural networks with asymmetric MFs based on the twice optimization algorithm for nonlinear system identification. *Inf. Sci.* **2023**, *929*, 123–143. [[CrossRef](#)]
43. Xu, D.; Zhang, S.; Zhang, H.; Mandic, D.P. Convergence of the RMSProp deep learning method with penalty for nonconvex optimization. *Neural Netw.* **2021**, *139*, 17–23. [[CrossRef](#)]
44. Barrau, A.; Bonnabel, S. Three examples of the stability properties of the invariant extended Kalman filter. *IFAC-PapersOnLine* **2017**, *50*, 431–437. [[CrossRef](#)]
45. Vladov, S.; Yakovliev, R.; Bulakh, M.; Vysotska, V. Neural Network Approximation of Helicopter Turboshaft Engine Parameters for Improved Efficiency. *Energies* **2024**, *17*, 2233. [[CrossRef](#)]
46. Vladov, S.; Petchenko, M.; Shmelov, Y.; Drozdova, S.; Yakovliev, R. Helicopters Turboshaft Engines Parameters Identification at Flight Modes Using Neural Networks. In Proceedings of the 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, 10–12 November 2022. [[CrossRef](#)]
47. Vladov, S.; Scislo, L.; Sokurenko, V.; Muzychuk, O.; Vysotska, V.; Osadchy, S.; Sachenko, A. Neural Network Signals Integration from Thermogas-dynamic Parameters Sensors for Helicopters Turboshaft Engines at Flight Operation Conditions. *Sensors* **2024**, *24*, 4246. [[CrossRef](#)] [[PubMed](#)]
48. Corotto, F.S. Appendix C—The method attributed to Neyman and Pearson. In *Wise Use of Null Hypothesis Tests*; Academic Press: Cambridge, MA, USA, 2023; pp. 179–188. [[CrossRef](#)]
49. Motsnyi, F.V. Analysis of Nonparametric and Parametric Criteria for Statistical Hypotheses Testing. Chapter 1. Agreement Criteria of Pearson and Kolmogorov. *Stat. Ukr.* **2018**, *83*, 14–24. [[CrossRef](#)]
50. Babichev, S.; Krejci, J.; Bicanek, J.; Lytvynenko, V. Gene expression sequences clustering based on the internal and external clustering quality criteria. In Proceedings of the 2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, 5–8 September 2017. [[CrossRef](#)]
51. Hu, Z.; Kashyap, E.; Tyshchenko, O.K. GEOCLUS: A Fuzzy-Based Learning Algorithm for Clustering Expression Datasets. *Lect. Notes Data Eng. Commun. Technol.* **2022**, *134*, 337–349. [[CrossRef](#)]
52. Centorrino, V.; Bullo, F.; Russo, G. Modeling and contractivity of neural-synaptic networks with Hebbian learning. *Automatica* **2024**, *164*, 111636. [[CrossRef](#)]
53. Wang, Y.; Wang, Z. Model free adaptive fault-tolerant tracking control for a class of discrete-time systems. *Neurocomputing* **2020**, *412*, 143–151. [[CrossRef](#)]
54. Anfilets, S.; Bezobrazov, S.; Golovko, V.; Sachenko, A.; Komar, M.; Dolny, R.; Kasyanik, V.; Bykovyy, P.; Mikhno, E.; Osolinskyi, O. Deep multilayer neural network for predicting the winner of football matches. *Int. J. Comput.* **2020**, *19*, 70–77. [[CrossRef](#)]
55. Andriushchenko, K.; Rudyk, V.; Riabchenko, O.; Kachynska, M.; Marynenko, N.; Shergina, L.; Kovtun, V.; Tepliuk, M.; Zhemba, A.; Kuchai, O. Processes of managing information infrastructure of a digital enterprise in the framework of the «Industry 4.0» concept. *East.-Eur. J. Enterp. Technol.* **2019**, *1*, 60–72. [[CrossRef](#)]
56. Vladov, S.; Yakovliev, R.; Vysotska, V.; Nazarkevych, M.; Lytvyn, V. The Method of Restoring Lost Information from Sensors Based on Auto-Associative Neural Networks. *Appl. Syst. Innov.* **2024**, *7*, 53. [[CrossRef](#)]
57. Vladov, S.; Yakovliev, R.; Hubachov, O.; Rud, J.; Drozdova, S.; Perekrest, A. Modified Discrete Neural Network PID Controller for Controlling the Helicopters Turboshaft Engines Free Turbine Speed. In Proceedings of the 2023 IEEE 5th International Conference on Modern Electrical and Energy System (MEES), Kremenchuk, Ukraine, 27–30 September 2023; pp. 797–802. [[CrossRef](#)]
58. Kovtun, V.; Altameem, T.; Al-Maitah, M.; Kempa, W. Simple Statistical Tests Selection Based Parallel Computing Method Ensures the Guaranteed Global Extremum Identification. *J. King Saud Univ.-Sci.* **2024**, *36*, 103165. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.