UDC: 004.89 https://doi.org/10.15407/jai2025.02.105

N. Lazarieva
Kharkiv National University of Radio Electronics, Ukraine
14 Nauky Avenue, Kharkiv, 61166
laznata@ukr.net
https://orcid.org/0000-0002-2387-6334

NEURAL NETWORK IMPLEMENTATION OF HIERARCHICAL FUZZY MODEL OF DYNAMIC OBJECTS SPEED CONTROL

Abstract. The aim is to create an intelligent control system based on soft computing for controlling dynamic objects moving along one of the defined routes in real-time systems. The parameters of objects in the real environment are characterized by high nonlinearity, dependence on the state of the environment, and time-varying dynamics when some parameters and states of objects are not available for measurement. Taking this into account, the hierarchical structure of the system is developed based on the classical fuzzy algorithms of Mamdani and Takagi-Sugeno-Kang, and an adaptive fuzzy neural network that implements the model. The application of a neuro-fuzzy model to controlling the movement of dynamic objects with many parameters and incomplete certainty through the use of expert knowledge is considered. A mathematical description of the fuzzy hierarchical model, a learning algorithm, and computer modeling are presented on the example of controlling the speed of rolling cars from a sorting hill. An example of the application of fuzzy rules built on numerical data is considered. The results of modeling with visualization of the results for the synthesized data are presented. The scientific innovation of the obtained results lies in the development of a hierarchical neuro-fuzzy model designed for forecasting and controlling dynamic objects. The modeling results confirm the ability of the proposed model to predict the unknown mapping of the input data vector, which consists of measured and unmeasured parameters, into the desired numerical value at certain points of the path at the model output. The obtained results demonstrate effective prediction of motion dynamics, the ability to achieve high forecasting accuracy and the possibility of intellectualizing the control of the technological process. When approximating the nonlinear dependence, the use of a multilayer neural network ensures the adaptability of the model to a specific area of application, and synergy with a fuzzy algorithm allows automating the process of controlling the technological process at the level of a human operator.

Keywords: adaptive control, fuzzy neural networks, hierarchical structure, Sugeno knowledge base, TSK algorithm, Mamdani algorithm, ANFIS, M-ANFIS.

Introduction

Automated control of objects and technological processes is often performed by a human operator based on instructions and his own experience. The high dynamics of the process, nonlinearity and dependence of parameters on environmental conditions is a problem that has not been successfully solved for many years. The application of classical control theory is unacceptable, and the use of mathematical methods has limitations due to the complexity of the dynamic process. The incompleteness of information and the impossibility of accurately measuring the parameters of the object, the diversity of environmental states, and the presence of disturbances are difficult to take into account. An infinite set of parameter combinations limits the application of statistical methods.

Control of the technological process of rolling wagons is a typical example of situational control. Control is successfully performed by a human operator without using any formalization of the object by mathematical methods. Therefore, the idea of using the experience of a human operator in an automated control system is the basis for its intellectualization.

The application of intellectual methods in unique operating conditions of complex systems has yielded positive results. There are many possibilities for combining artificial intelligence technologies to obtain hybrid algorithms that use uncertain and imprecise data and generate approximate solutions that are not inferior to a human operator. By eliminating the need to define a mathematical model of the object, the concept of fuzzy control finds wide application, allowing to build control systems for technological processes in conditions of incomplete certainty, which is characteristic of real processes. The most effective is the use of soft computing based on fuzzy algorithms for systems of high complexity with the use of neural network adaptation. At the same time, hybrid models require a relatively short development time and are quite reliable.

Problem statement

With an unknown object control law, the greatest difficulty in applying soft computing lies in constructing fuzzy rules and selecting model parameters, especially with the overall complexity of the object and a large number of influencing factors. The use of neural networks allows to simulate human intellectual ability, and hybrid models built on their basis are able to adapt the model according to the application area and work with an unknown control law.

The aim of the research

The aim of the research is to create an adaptive hierarchical fuzzy neuron network for controlling complex dynamic objects and to applicate setting methods to implement an unknown mapping of a set of parameters at the input to the control signal

$$y^{(k)} = f(x_1^{(k)}, x_2^{(k)}, ..., x_n^{(k)}) .$$

Mathematical formulation of the problem:

– X is a set of descriptions of the object and the environment that forms the input vector;

– Y is a set of predicted solutions.

The unknown target dependence is the mapping $X \to Y$. It is necessary to build a prediction algorithm for an arbitrary vector of parameters in an infinite set of states and implement the model.

Analysis of research and publications

Modeling with fuzzy knowledge bases, proposed by Takagi T. and Sugeno M. [1], has found wide practical application in control systems [2]. However, fuzzy systems have a number of disadvantages, among which [3, 4]:

– lack of standard methods for transferring knowledge to the rule base;

– incompleteness or contradiction of the set of fuzzy rules;

– discrepancy of selected membership functions and their parameters;

– need for effective setting methods of model parameters for specific application conditions.

The ability to adjust the parameters of membership functions and extract rules during the learning process makes it possible to eliminate these disadvantages.

Considering the world experience of building intelligent systems, it can be noted that various approaches have been developed for the effective combination of different methods in order to simplify model building and optimize parameters. Research on hybrid algorithms has yielded good results and wide practical application. The ability to change internal parameters to work in certain conditions, which is an undeniable advantage of using neural networks in building a model, was studied by Takagi and Hayashi [5]. The authors [6] showed that the functional behavior of radial basis function networks and fuzzy inference systems is actually equivalent. They have common characteristics in data operations and learning process to achieve the desired mapping of input variables into the output signal, and the learning rules can be applied to both models.

Hybrid algorithms significantly simplify the creation of membership functions, provide adaptation to environmental changes and enable the generation of fuzzy logical control decisions with a given accuracy.

The implementation of fuzzy logic by multilayer neural networks, initiated in [7], contributed to the creation of effective automatic process control systems in many areas. The most widely used is the Adaptive Neuro-Fuzzy Inference System (ANFIS) [8]. The model presented in [3] is based on the Takagi-Sugeno fuzzy inference algorithm for approximating nonlinear functions. The set of linear equations allows setting parameters by classical optimization methods [9]. After direct propagation, estimation is performed by the Least Squares Method (LSE) and training with backpropagation of the error by the Gradient Descent Method [10]. Almost all nonlinear dynamical systems can be represented by Takagi-Sugeno fuzzy models with a high degree of accuracy [11]. The advantages of such a fuzzy structure include the simplicity of calculating the output data [12].

A widely used algorithm for fuzzy reasoning is also the Mamdani fuzzy inference system. The implementation of this algorithm by an adaptive neural network is considered in [4] and the structure of the M-ANFIS model is proposed. The method of training the Mamdani fuzzy logical inference system based on artificial neural networks gives good results.

Thus, the summed effect of using various intelligent methods allows obtaining hybrid systems, the capabilities of which exceed the capabilities of the individual methods. This allows to build models that adapt to a changing environment [13], which is especially important for open systems of high complexity.

Presentation of the main research material

In control tasks, the membership functions of fuzzy models and the control result can be obtained by an expert method, but the way to achieve the required result is almost always unknown. Therefore, the identification of the situation for decision-making consists in identifying the interdependence between the required control and the factors characterizing the object and the environment.

The accuracy of the model is limited by the depth of the input data, as a set of parameters that affect the result. Since the rolling control system is not isolated, to ensure high accuracy, direct or indirect factors affecting the dynamics of the object must be taken into account as much as possible. The speed of movement depends on the action of external forces acting on a moving object and on the inertia of the object itself, which depends on the mass, number of axes or number of units.

To achieve the set goal of the research, it is necessary to perform structural and parametric identification of the model, namely:

– based on the analysis of the functional elements of the fuzzy model, determine the structure of an adaptive neural network that implements a fuzzy control algorithm;

– build and train a neuro-fuzzy network that implements an unknown mapping of a set of parameters from sensors and linguistic information from experts into an output signal.

Structural model identification

With increasing system complexity, it leads to a significant increase in data processing time, making the method impractical for real-time operation [13]. Therefore, having multi-parameter data and taking into account the need to use them with a significantly smaller number of inputs, a hierarchical structure construction is used [14], where the first level reduces the dimensionality of the input vector based on the Mamdani algorithm [4, 15], and the second level performs the logical output of the control signal using the Takagi-Sugeno fuzzy reasoning algorithm [1, 3, 8].

The application of the principle of hierarchy when building a knowledge base of a fuzzy system aims to reduce the dimensionality of the input data and allows adding new variables with further increase in knowledge about the object or environment.

The sequence of developing a control solution:

1. Obtaining current data.
2. Granulation of input data with obtaining integral indicators:
   – evaluation of information features of motion dynamics (Vi, Vp, Av);
   – evaluation of information features of the object (Vw, Nv, No);
   – evaluation of information features of the route (Nk, Ks, Kk);
   – evaluation of information features of the environment (T, B, Vv, Op).
3. Fuzzy logical inference based on integral indicators Object, Move, Route, Environ.

The first-level inference mechanism of a fuzzy control model combines certain input parameters into a composite output variable. One of the common measures is the multiplicative convolution of certain data components when assessing the importance of an integral indicator. By using the Mamdani for n_x variables x_i, $i = 1, 2, ..., n_x$, rule base with the implementation of the T-norm product, the convolution of membership functions as signs of the importance of indicators in the rule is performed. This linguistically expresses the dependence of the composite indicator on the set of parameters of the underlying reality:

$$R^{(k)}: IF\ x_1\ IS\ A_{j1}^{(k)}\ AND\ ...\ x_{n_x}\ IS\ A_{jn_x}^{(k)}\ THEN\ y\ IS\ B_j^{(k)}$$

The second-level inference mechanism differs in the type of rules used. Modeling based on an Adaptive Neuro-Fuzzy Inference System according to the Takagi-Sugeno-Kang algorithm [1] implements the unknown mapping $f:(y_1, y_2, ..., y_{n_y}) \to z$, where z is the output value of the fuzzy system, the control signal. The input variables are information granules y_i, obtained in the form of a vector of integral data $Y = \{y_1, y_2, ..., y_{n_y}\}$, where n_y is the number of second-level input variables

$$Y = f(\mu_{A_1}(x_1), \mu_{A_2}(x_2), ..., \mu_{A_{n_x}}(x_{n_x})).$$

The functioning is described by the rule base $R^{(k)}$, which linguistically expresses the dependence of the composite indicators on the values of the components

$$R^{(k)}: IF\ y_1\ IS\ B_1^{(k)}\ AND ...\ y_{n_y}\ IS\ B_{n_y}^{(k)}$$

$$THEN\ z^{(k)} = f^{(k)}(y_1, y_2, ..., y,$$
$$z^{(k)} = q_0^{(k)} + q_1^{(k)} y_1 + ... + q_{n_y}^{(k)} y_{n_y}.$$

The operators of the fuzzy rule base are implemented by next functions:

– algebraic product for the operator "AND" and implications $X \to Y$;

– algebraic sum for the operator "OR" and aggregation of the consequences of the rules;

– centroid for defuzzification of the value $\mu_B(y)$ of the integral variable y_i.

The form of representation of fuzzy sets is the generalized Gaussian function

$$\mu = \frac{1}{1 + \left(\frac{x-c}{a}\right)^{2b}}.$$

The parameters that are adjusted during the training process are:

– a (standard deviation);

– b (stretching coefficient);

– c (maximum coordinate).

With n input variables, the number of nonlinear parameters of the conditional part is $3n$.

The main steps of fuzzy inference are implemented in successive layers of the network. The choice of specific parameters, such as: network layers, layer components and connections are determined according to the functional organization of the fuzzy algorithm.

For a hierarchical network, the general functional dependence for N integral indicators with parallel processing in local modules is written as the expression

$$z(Y) = \prod_{p=1}^{N} \left[\frac{\sum_{j=1}^{k} \left(\prod_{i=1}^{n_x} \mu_{A_i}(x_i) \cdot \mu_{B_i} \right)}{\sum_{j=1}^{k} \mu_{A_{ij}}} \right] \times$$
$$\times \frac{\sum_{j=1}^{k} \left(\prod_{i=1}^{n_y} \mu_{B_i}(y_i) \right) \left(q_{0j} + \sum_{i=1}^{n_y} q_{ij} y_i \right)}{\sum_{j=1}^{k} \left(\prod_{i=1}^{n_y} \mu_{B_i}(y_i) \right)}.$$

The corresponding structure for the implementation of a hierarchical fuzzy control model is shown in Fig. 1.

The structure corresponds to a multilayer neural network of direct signal propagation, which is the most common implementation of fuzzy algorithms.

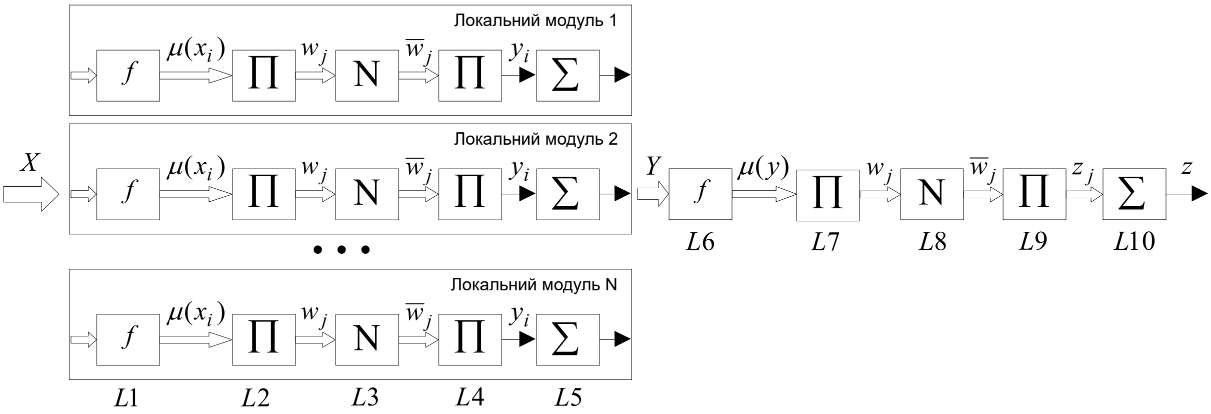

Fig. 1. Structure of a fuzzy control model

Neural network implementation

The network has several inputs, represented by numerical and linguistic variables, and a single output. Each neuron, as an element of a layer, performs a certain function on the incoming data according to the fuzzy algorithm.

Layer L1: Input vector $X = \{x_1\ x_2\ ...\ x_{n_x}\}$ is an array of dimension $1 \times n_x$. The signal at the output of the i-node is defined as $y_i^{[1]} = f(x_i)$, where f is a defined membership function with a set of nonlinear parameters that are adapted during parametric identification.

The output of the first layer is a membership degree matrix of dimension $m_A \times n_x$. The number of nonlinear parameters of the rules antecedent is equal to $3m_A n_x$.

Layer L2: Determining the degree of truth of rule antecedents.

The output signal of the i-node is the weight coefficient of the j-rule for the current vector X of input variables. The nodes of this layer implement the T-norm by the product

$$y_j^{[2]} = \mu_{A_1}(x_1) \cdot ... \cdot \mu_{A_{n_x}}(x_{n_x}) = w_j .$$

The number of nodes is determined by the number of rules k.

At the output of this layer, the vector of the strength of the conditional part of the rules is of dimension $k \times 1$.

Layer L3: Implication with determination of the rule consequences. The output value of nodes is determined by the product

$$y_j^{[3]} = w_j \cdot \mu_{B_i} .$$

The number of nodes of this layer can be calculated as km_y. The number of adaptive nonlinear parameters depends on the number of fuzzy terms of consequent membership function as $3m_B k$.

Layer L4: Aggregation and determination of the fuzzy output signal as the sum of the values, coming from the previous layer. The value at the output of the nodes

$$y^{[4]} = \sum_{j=1}^{k} y_j^{[3]} = \sum_{j=1}^{k} w_j \cdot \mu_{B_i} .$$

Layer L5: Determination of the integral parameter of the local module by the Center Of Gravity method. The value at the output of the node

$$y^{[5]} = \frac{\sum_{j=1}^{k} y_j^{[4]}}{\sum_{j=1}^{k} \mu_{B_{ij}}} = \frac{\sum_{j=1}^{k} w_j \cdot \mu_{B_{ij}}}{\sum_{j=1}^{k} \mu_{B_{ij}}} .$$

Layer L6: Determination of membership degree of the current vector $Y = \{y_1\ y_2\ ...\ y_{n_y}\}$ to the fuzzy set $\mu_{B_i}(y_i)$. The value at the output is determined by the defined function $z_i^{[6]} = f(y_i)$ with the formation of a matrix of degrees of membership at the output of this layer.

The set of nonlinear parameters a, b, c, which are adapted during training for n_y

integral variables is equal to $3m_B n_y$.

Layer L7: Determination of the strength of the j-rule w_j for the vector Y. When modeling the T-norm by the product operation at the output of each node

$$z^{[7]} = \mu_{B_i}(y_1)\cdot\ldots\cdot\mu_{B_{n_y}}(y_{n_y}) = w_j$$

The number of nodes in this layer is equal to the number of fuzzy rules.

Layer L8: Calculation of the ratio of the strength of individual rules by normalizing the strength of the j-rule

$$z^{[8]} = \frac{w_j}{w_1 + w_2 + \ldots + w_k} = \overline{w}_j,$$
$$j = 1,2,\ldots,k.$$

The output is a vector of normalized rule strength with dimension $k \times 1$.

Layer L9: Calculation of the consequence j-rules. The nodes of this layer determine the product of the normalized value of the rule strength $\overline{w}_j$ and the linear function of the consequent part of the j-rule

$$z^{[9]} = \overline{w}_j \cdot f^{(k)} = z_j,$$
$$f^{(k)} = q_0 + \sum_{i=1}^{n_y} q_i y_i,\ i = 1,2,\ldots,n_y.$$

The parameters that are adapted are linear weight coefficients q_i, the number of which is $k(n_y + 1)$.

Layer L10: Aggregation of the results of individual rules with calculation of the general output value

$$z^{[10]} = \sum_{j=1}^{k} z_j^{[9]},\ j = 1,2,\ldots,k,$$
$$\hat{z}(Y) = \sum_{j=1}^{k} \overline{w}_j \cdot \left(q_0 + \sum_{i=1}^{n_y} q_i y_i \right).$$

Adaptive neuro-fuzzy network has: 10 layers;

$$N \cdot (n_x m_x + 2k_x + 2) + (n_y m_y + 3k_y + 1)$$

nodes;

$$3 \cdot (m_A n_x + m_B k_x + m_C n_y) + k_y (n_y + 1)$$

adaptive parameters.

Parametric identification

The backpropagation algorithm optimizes the parameters of the membership functions and linear coefficients of the consequents to achieve the desired mapping of the input data vector into the output signal. The classical training procedure involves presenting a training set of input-output patterns. The loss function is defined as the sum of the differences between the predicted $\hat{z}(Y_j)$ and the target value z_j for each data set $\varepsilon = (\hat{z} - z)$.

The sought value is a function $\hat{z} = f(a,b,c,\ q_0,q_1,..,q_n)$ determined by a set of model parameters. Based on the Least Square Error (LSE) method, the values of the network parameters that minimize the loss function are found

$$E = \frac{1}{2}\sum_{j=1}^{k}\left(z_j - \hat{z}(Y)\right)^2 \to \min.$$

Layers L9, L6, L4, L1 are subject to training with parameter adaptation. According to the gradient descent procedure, the coefficients of the consequents $[q_0^k, q_1^k, \ldots, q_n^k]$ and the parameters of the bell functions $[a^k, b^k, c^k]$ of the rule antecedents for the integral data $Y = \{y_1, y_2, \ldots, y_n\}$ are updated at L2.

The following parameters are updated at level 1: the parameters of the bell functions $[a^k, b^k, c^k]$ of the consequents and the fuzzy terms of the antecedents of the rules.

Layer L10: Determining the rate of change of the error as a derivative of the loss function

$$\frac{\partial E}{\partial \hat{z}} = \varepsilon.$$

Layer L9: Determining the gradients of the parameters of the rule consequents of L2

$$\hat{z}_j = f(q_1, q_2, \ldots, q_k),$$

$$\Delta q_i = -\eta \frac{\partial E}{\partial \hat{z}} \cdot \frac{\partial \hat{z}}{\partial q_i}.$$

Layer L6: Calculating the gradients on the nonlinear parameters of the generalized Gaussian function of the rule antecedents $w = f(a,b,c)$

$$\Delta a^{[6]} = -\eta \frac{\partial E}{\partial \hat{z}} \cdot \frac{\partial \hat{z}}{\partial w} \cdot \frac{\partial w}{\partial \mu} \cdot \frac{\partial \mu}{\partial a},$$

$$\Delta b^{[6]} = -\eta \frac{\partial E}{\partial \hat{z}} \cdot \frac{\partial \hat{z}}{\partial w} \cdot \frac{\partial w}{\partial \mu} \cdot \frac{\partial \mu}{\partial b},$$

$$\Delta c^{[6]} = -\eta \frac{\partial E}{\partial \hat{z}} \cdot \frac{\partial \hat{z}}{\partial w} \cdot \frac{\partial w}{\partial \mu} \cdot \frac{\partial \mu}{\partial c}.$$

Layer L4: Determining the partial derivatives of the parameters of the consequents of L1 on the baze of $Y = \{y_1, y_2, ..., y_n\}$

$$\Delta a^{[4]} = -\eta \frac{\partial E}{\partial \hat{z}} \cdot \frac{\partial \hat{z}}{\partial y_i} \cdot \frac{\partial y_i}{\partial \mu} \cdot \frac{\partial \mu}{\partial a},$$

$$\Delta b^{[4]} = -\eta \frac{\partial E}{\partial \hat{z}} \cdot \frac{\partial \hat{z}}{\partial y_i} \cdot \frac{\partial y_i}{\partial \mu} \cdot \frac{\partial \mu}{\partial b},$$

$$\Delta c^{[4]} = -\eta \frac{\partial E}{\partial \hat{z}} \cdot \frac{\partial \hat{z}}{\partial y_i} \cdot \frac{\partial y_i}{\partial \mu} \cdot \frac{\partial \mu}{\partial c}.$$

Layer L1: Determining the partial derivatives for the parameters of the membership functions of the antecedents of L1.

$$\Delta a^{[1]} = -\eta \frac{\partial E}{\partial \hat{z}} \cdot \frac{\partial \hat{z}}{\partial y_i} \cdot \frac{\partial y_i}{\partial w_j} \cdot \frac{\partial w_j}{\partial \mu} \cdot \frac{\partial \mu}{\partial a},$$

$$\Delta b^{[1]} = -\eta \frac{\partial E}{\partial \hat{z}} \cdot \frac{\partial \hat{z}}{\partial y_i} \cdot \frac{\partial y_i}{\partial w_j} \cdot \frac{\partial w_j}{\partial \mu} \cdot \frac{\partial \mu}{\partial b},$$

$$\Delta c^{[1]} = -\eta \frac{\partial E}{\partial \hat{z}} \cdot \frac{\partial \hat{z}}{\partial y_i} \cdot \frac{\partial y_i}{\partial w_j} \cdot \frac{\partial w_j}{\partial \mu} \cdot \frac{\partial \mu}{\partial c}.$$

The parameters are updated after each training example according to the stochastic gradient descent procedure

$$q_i^k(n+1) = q_i^k(n) - \eta \frac{\partial E}{\partial q_i^k},$$

$$a_i^k(n+1) = a_i^k(n) - \eta \frac{\partial E}{\partial a_i^k},$$

$$b_i^k(n+1) = b_i^k(n) - \eta \frac{\partial E}{\partial b_i^k},$$

$$c_i^k(n+1) = c_i^k(n) - \eta \frac{\partial E}{\partial c_i^k}.$$

Implementation example

Regulating the speed of wagons rolling under the action of many forces refers to intellectual tasks with an unknown algorithmic control method.. The solution is understood as the execution of actions necessary to achieve the control goal – maintaining safe distances during movement and approaching a given point at an allowable speed.

The surface that characterizes the state of the object under certain conditions is described by an unknown function in multidimensional space. For an object with the parameters [16] given in Table 1, the boundaries of the variable carriers, the number of terms and the initial fuzzy control rules based on numerical data are determined.

Table 1. Tuples of arbitrary input data features and the reference output value

Vi	1,4	1,2	1,3	1,4	1,2	1,4	1,3	1,4
Vw	22	162	252	500	82	612	282	22
Nv	1	2	7	20	1	12	3	1
No	4	8	40	80	8	48	12	4
T	-5	-25	-5	15	-20	18	-1	-5
B	70	180	45	30	130	60	0	70
Vv	2	2,7	0,8	4	2,7	10	5	2
Point	2,8760	4,7429	5,397	5,1906	4,3668	5,4512	3,4551	5,7994

For the input set of parameters $n_x = 7$, the architecture of the fuzzy neural network is presented in Figure 2.

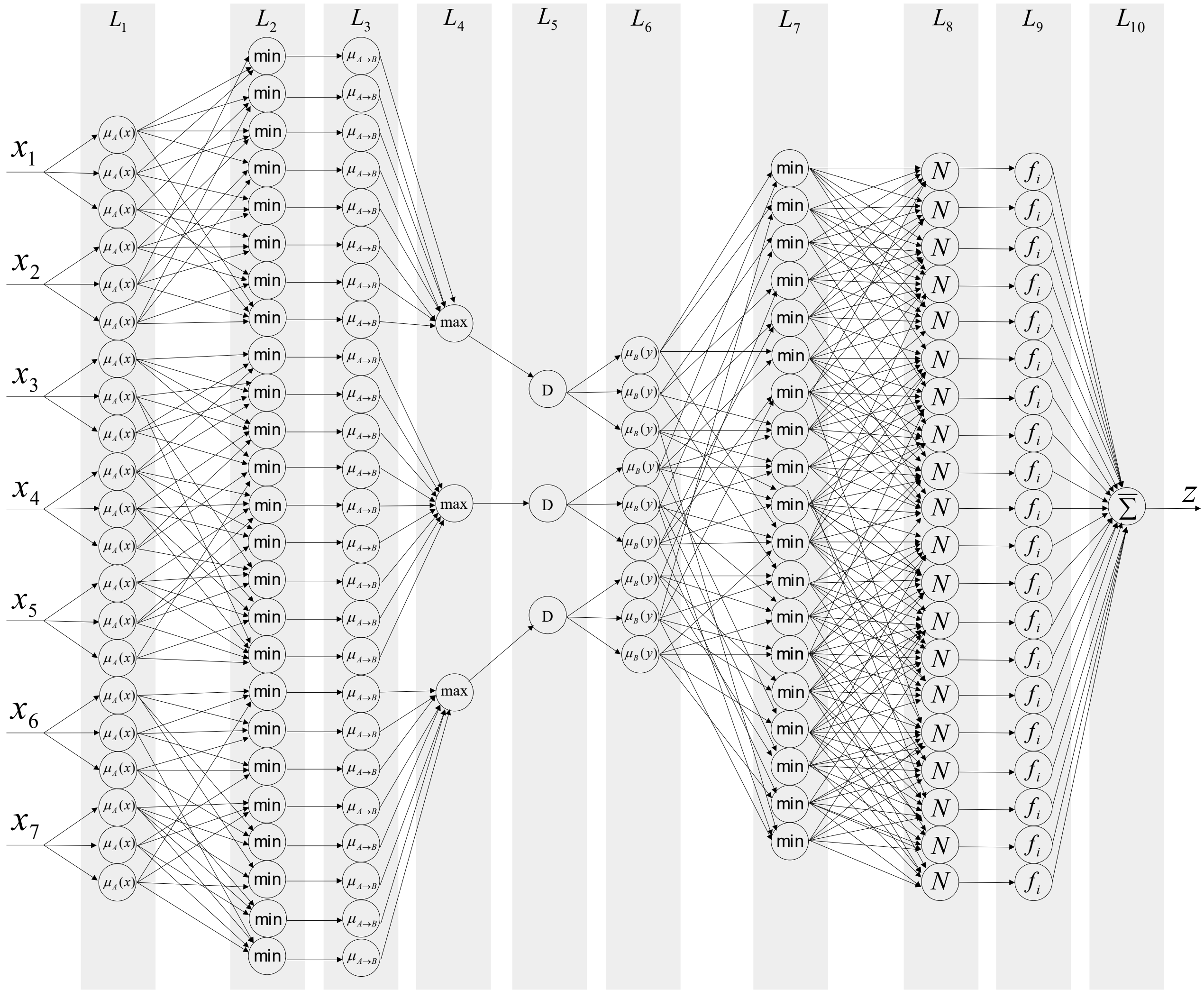

Fig. 2. Architecture of a hierarchical neuro-fuzzy adaptive network

It is necessary to obtain the optimal output signal of a fuzzy system based on a sequential algorithm for processing data in local modules and combining generalized variables.

To simulate a control system of moving object, a Python program was developed that implements a fuzzy algorithm and visualizes the results of model training. Figure 3 shows the change in prediction error during the training process.

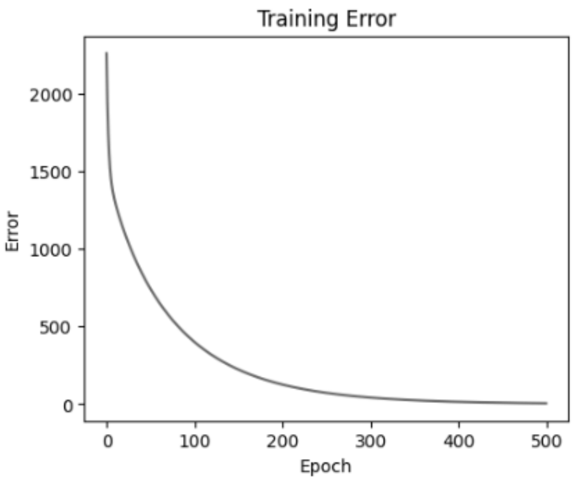

Fig. 3. Training Error

Prediction Error after 500 training epochs:

– MSE: 1,533040;
– LSE: 6,132160.

The neural network is trained to identify nonlinear subspace boundaries in the conditional part and nonlinear equations in the consequent part of fuzzy rules. An example of optimized consequent parameters is shown in Figure 4.

Consequent Parameters:

	Rule	p0	p1	p2	p3	p4	p5	p6	p7
0	Rule 0	0.09952	0.11898	0.00473	-0.00955	0.06990	-0.04473	-0.04916	-0.05564
1	Rule 1	0.00196	0.02370	0.00603	0.04769	0.04965	-0.10378	0.05015	-0.02167
2	Rule 2	0.02877	-0.10320	0.00887	-0.02233	0.07404	0.04902	-0.00905	-0.10502
3	Rule 3	-0.04771	0.08610	0.00193	0.10197	0.01266	-0.07085	-0.08884	0.00762
4	Rule 4	0.07399	0.02263	0.04855	0.21756	-0.00983	0.05446	-0.08556	0.11169
5	Rule 5	0.00940	0.04642	-0.01042	0.01250	-0.28073	-0.01489	0.08647	-0.11976
6	Rule 6	0.04966	0.01777	-0.00477	0.07809	-0.02918	0.02819	0.04086	0.00785
7	Rule 7	0.00523	-0.03270	-0.02550	-0.01123	-0.01456	0.02686	0.07035	0.00922

Fig. 4. Fragment of the parameter table of consequences of fuzzy rules

A set of modified parameters determines the behavior of the constructed fuzzy network. Figure 5 shows a comparison of the prediction result and the reference value of the object state.

Target vs Predicted Values:

	Target	Predicted
0	5.62853	4.05504
1	5.18197	5.18199
2	4.71716	4.67393
3	4.57550	4.57550
4	5.35840	5.35235
5	3.40730	4.62282
6	5.01322	5.89763
7	4.84945	2.10594

Fig 5. Fragment of test results

The response surface shown in Figure 6 characterizes the influence of the environment, in particular the direction and speed of the wind, on the control signal.

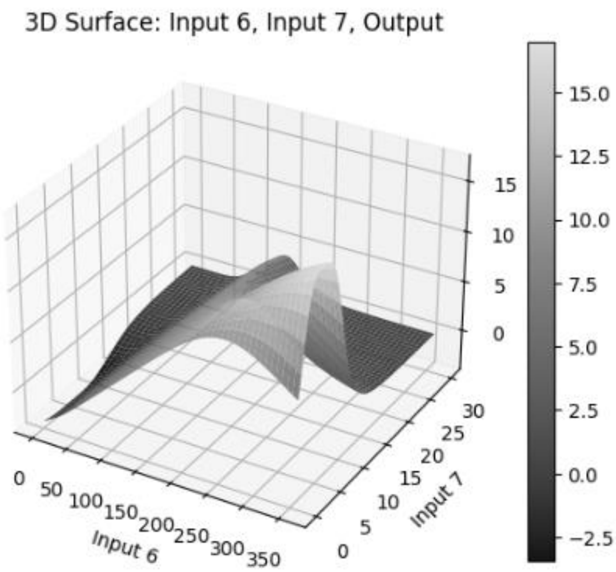

Fig. 6. The influence of the environment on the control signal.

In the testing mode of the trained model, the predictions were checked on an arbitrary example:

– predicted result = 4.67434;
– actual target = 4.71716.

The operation of individual terms is shown in Figure 7.

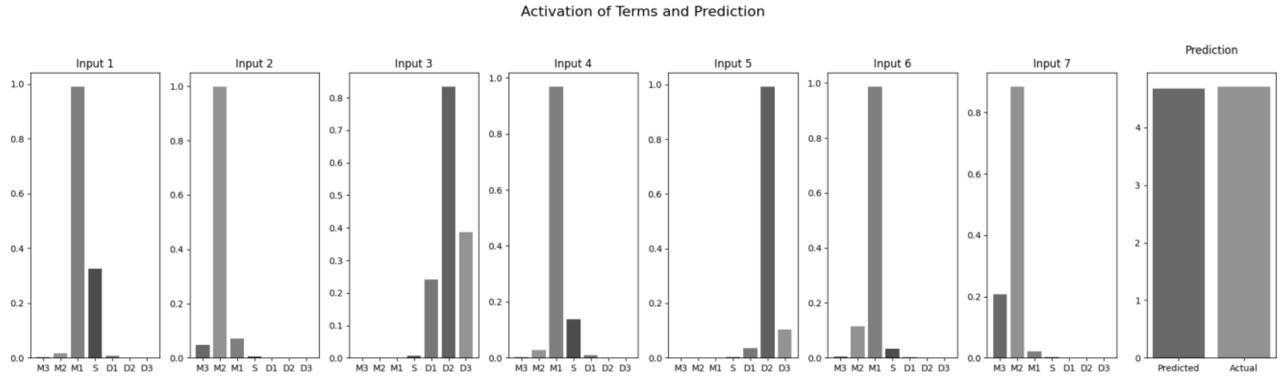

Fig. 7. Activation of terms and prediction

Conclusions

The result of the research is the creation of an adaptive hierarchical fuzzy neural network. Expressions for determining the output of each layer, the total number of nonlinear adaptive parameters are presented. The program in Python for processing data by three channels and inference by the TSK algorithm with visualization is developed. The modeling results show that the fuzzy algorithm implemented by a neural network with learning and adaptation of model parameters provides the implementation of control of dynamic objects specified in the n-dimensional state space. The hierarchical structure makes it possible to take into account the operating conditions that determine the change in the characteristics of the studied objects.

Regarding the effectiveness of the proposed approach, the neuro-fuzzy hierarchical model is created in accordance with a certain concept of controlling the movement of wagons, has both advantages in terms of the goal, and certain limitations and disadvantages. Under the influence of a large number of factors that are unmeasurable, unknown and act in different directions, many real-world objects function. Predicting the behavior of such objects and its targeted correction during operation allows to improve the quality of control, which is relevant for many processes.

Further improvement may be aimed at integrating an Application Programming Interface for interaction with external programs and obtaining data on the meteorological situation to expand knowledge about the operating conditions of the control system.

References

1. Takagi T. & Sugeno M. (1985) Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. on Systems, Man, and Cybernetics*, 15:116–132.

2. Sugeno M., editor. (1985) Industrial applications of fuzzy control. *Elsevier Science Pub*. Co.

3. Jang, J.S.R. (1993) ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems Man & Cybernetics*, 23, 665-685. https://doi.org/10.1109/21.256541.

4. Chai Y., Jia L., Zhang Z. (2009) Mamdani model based adaptive neural fuzzy inference system and its application. *International Journal of Computer, Electrical, Automation, Control and Information Engineering* 3(3).

5. Takagi H. & Hayashi I. (1988) Artificial neural network-driven fuzzy reasoning, *Proc. Int. Workshop Fuzzy System Applications* (IIZUKA88), Iizuka, Japan, 217C218.

6. Jang, J-S & Sun, C-T. (1993). Functional Equivalence between Radial Basis Function Networks and Fuzzy Inference Systems. *IEEE Transaction on Neural Networks*, 4(1), 156-159.

7. Gite, A.V., Bodade, R.M., & Raut, B.M. (2013). ANFIS controller and its application. *International Journal of Engineering Research and Technology*, 2(2).

8. Abraham, A. (2005). Adaptation of fuzzy inference system using neural learning. *In Fuzzy Systems Engineering: Theory and Practice,* pp. 53-83. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/11339366_3, ISBN 978-3-540-25322-8.

9. Riza L.S., Bergmeir C., Herrera F., & Benítez J.M. (2015). frbs: Fuzzy rule-based systems for classification and regression in R. *Journal of statistical software*, 65, 1-30.

10. Raja P., & Pahat B. (2016). A review of training methods of ANFIS for applications in business and economics. *International Journal of u-and e-Service, Science Technology*, 9(7), 165-172.

11. Mehran K. (2008). Takagi-sugeno fuzzy modeling for process control. *Industrial Automation, Robotics and Artificial Intelligence* (EEE8005), 262, 1-31.

12. Schnitman L., & Yoneyama T. (2001). Takagi-Sugeno-Kang fuzzy structures in dynamic system modeling. *In Proceedings of the IASTED International Conference on Control and Application* (CA'2001), 160-165.

13. Jang J.S.R., Sun C.T., & Mizutani E. (1997). Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence [Book Review]. *IEEE Transactions on automatic control*, 42(10), 1482-1484.

14. Lazarieva N. (2023) Modeling of multi-factory dependences in complex control systems by Sugeno fuzzy knowledge base. *Artificial Intelligence* 1, 138-146.

15. Mamdani E.H. (1977) Application of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE Transactions on Computers*, C-26(12), 1182-1191, doi: 10.1109/TC.1977.1674779.

16. Lazarieva N.M., Lazariev O.V. (2023) Creating of a rule base according to numerical data for the fuzzy control system of retarders. *X Vseukrainska naukovo-praktychna konferentsiia zdobuvachiv vyshchoi osvity ta molodykh vchenykh z avtomatychnoho upravlinnia* Kherson-Khmelnytskyi: KhNTU, 78-81.

The article has been sent to the editors 27.05.25.
After processing 10.06.25.
Submitted for printing 30.06.25.

Copyright under license CCBY-SA4.0.