CONSTRUCTION OF AN ANALYTICAL METHOD FOR LIMITING THE COMPLEXITY OF NEURAL-FUZZY MODELS WITH GUARANTEED ACCURACY

B. Sytnik
PhD, Associate Professor*
E-mail: sitnik@kart.edu.ua

V. Bryksin
PhD, Associate Professor*
E-mail: vladimir.bryksin@gmail.com

S. Yatsko
PhD, Associate Professor**
E-mail: ua.yatsko@gmail.com

Y. Vashchenko
PhD, Senior Lecturer**
E-mail: yaroslav.vashchenko@gmail.com

*Department of Information Technologies***
**Department of Electrical Engineering and Electromechanics***
***Ukrainian State University of Railway Transport
Feierbach sq., 7, Kharkiv, Ukraine, 61050

1. Introduction

It is known that movable technical objects with nonlinear and variable parameters (for example, vehicles, robots, drones) functioning under the influence of disturbances, depending on weight, wind, trajectory, and profile, the obstacles of varying intensity [1]. To execute control over them, it is necessary to nonlinearly correct the parameters in the settings of control system, to form the assigned motion trajectories, to limit the required motion speed in a speed control setpoint, etc. [2].

The use of conventional digital PID-controllers in this case turns out to be not effective enough, given a significant number of random, chaotic, non-linear and fuzzy variable parameters in the characteristic equations of a closed system [3].

To set the desired nonlinear motion trajectories of movable objects and to nonlinearly correct the parameters for setting the controllers in control systems, a guaranteed accuracy in the implementation of their models is needed [4].

In cases when the application of conventional methods to control the system proves to be not efficient enough due to a lack of precise knowledge about the object of control, specialized methods of calculation and decision making are used. These algorithms include adaptive technologies, fuzzy logic, artificial neural networks, genetic algorithms.

2. Literature review and problem statement

In [5], authors derived a simplified analytical assessment of the minimally required number of approximation sections in the curve of motion trajectory of second order by the trapezoidal method with guaranteed accuracy. Such an estimate is
missing for curves of higher orders [6]. In order for the law of control to account for all variables that affect the process of control, the adaptive [7], situational [8], fuzzy [9], neural and neural-fuzzy model and regulators [10], as well as genetic algorithms [11] are used. The specified algorithms can improve the properties of control and decision-making systems; however, the efficiency of their functioning is defined by the optimal structure capable of solving the predefined class of tasks. The effectiveness of methods for neural-fuzzy modeling and control improves if they are used in combination with methods based on intelligent machines [12].

An analysis of papers in this field [13–15] has revealed a lack of analytical dependences for determining the complexity of models depending on the assigned accuracy of their implementation.

Paper [13], for example, is based on the research from psychology on that a person with average capabilities can simultaneously keep in memory from 5 to 9 information granules (terms). Thus, when implementing a fuzzy production model, the number of terms for the input and output variables should be chosen from 5 to 7. The number $r$ in production rules is proposed to be derived from formula:

$$r = z^m,$$

where $z$ is the number of fuzzy sets, $ω$ is the number of inputs of $x_i$ model.

The number of parameters $p$ for assigning the membership functions is recommended to choose from formula:

$$p = r + ω \cdot z = r^m + ω \cdot z,$$

and when implementing a model using ANN, the initial number of neurons $m$ in the intermediate layer of ANN is recommended to choose from formula:

$$m = \sqrt{pl},$$

where $p$ and $l$ are the numbers of input and output neurons, respectively.

However, a given recommendation does not make it possible to minimize the number of terms for fuzzy variables or the number of neurons in the intermediate layer of ANN depending on the required accuracy for model implementation.

In [14], the number of neurons $m$ in the intermediate layer of ANN is recommended to be adjusted until achieving the desired accuracy of approximation. For genetic algorithms [15], in the process of ANN learning, achieving the required accuracy of simulation in search procedures also requires the knowledge of the required number of chromosomes, mutations, distances between them, etc.

Thus, as noted, existing methods for determining the minimally required number of terms or neurons fail to provide for a guaranteed accuracy of model implementation.

The solution to a given problem can be obtained based on estimating the area of a model curve approaching the system one at the sections of function approximation.

Consider a model of the system that is described by a function of high order in the form:

$$y_c = f_c(x) = a_1x^{2l} + a_2x^{2l-1} + \ldots + a_lx + a_0,$$

that has $2l$ derivatives, for instance, for $2l \geq 2$:

$$y_c = \mu \sqrt{x_c},$$

$$y_c = A \sin(\mu x_c),$$

etc. (2)

An example of the chart $y_c = f_c(x)$ for system curve $y_c = \sqrt{x_c}$ and the model curves implemented by the method of rectangles (1) and the trapezoidal method (2) is shown in Fig. 1.

4. Model of the system and the development of a method for limiting the complexity of neural-fuzzy models with guaranteed accuracy

Consider a model of the system that is described by a function of high order in the form:

$$y_c = f_c(x) = a_1x^{2l} + a_2x^{2l-1} + \ldots + a_lx + a_0,$$ (1)

and when implementing a model using ANN, the initial number of neurons $m$ in the intermediate layer of ANN is recommended to choose from formula:

$$m = \sqrt{pl},$$

where $p$ and $l$ are the numbers of input and output neurons, respectively.

Thus, as noted, existing methods for determining the minimally required number of terms or neurons fail to provide for a guaranteed accuracy of model implementation.

The solution to a given problem can be obtained based on estimating the area of a model curve approaching the system one at the sections of function approximation.

3. The aim and objectives of the study

The aim of this study is to develop an analytical method for limiting the complexity of neural-fuzzy models that provide for a guaranteed accuracy of their implementation during approximation of functions. This must ensure a decrease in redundancy, as well as the rational selection of the number of terms for fuzzy variables or neurons depending on the assigned required accuracy.

To accomplish the aim, the following tasks have been set:

– to devise a procedure for estimating the required number of neurons (terms) in a model that provide for the necessary precision in approximating the area of a model curve to the system one at the sections of function approximation;

– to devise a procedure for estimating the required number of neurons (terms), which provide for the necessary accuracy of model implementation based on the maximum deviation between the system and model curves at the section of approximation;

– to consider practical implementation of constructing a fuzzy model and a model that uses ANN for a system with the assigned dependence.

4. 1. Estimation of the number of neurons (terms) such that a model curve approximates the system curve

To assess the required number of neurons (terms) in a model that provide for the necessary precision of approximating the area of a model curve to the system curve at the sections of function approximation, we estimated an error of...
approximation. An estimate of the approximation error can be obtained based on the residual terms of decomposition, in the Lagrange form, of the areas of system function \( f(x) \) into a Maclaurin series. The abscissa coordinate of the modeled system function \( x_c \) and the dimensionless relative abscissa coordinate of model function \( x \) are related via interrelation:

\[
\frac{(x-0)}{(1-0)} = \frac{(x-a)}{(b-a)},
\]

(3)

hence:

\[
n\Delta_c = nh = n\frac{(x_c-a)}{(b-a)} = n\Delta_c = 1.
\]

(4)

To calculate the area under curve \( f(x) \) at the assigned accuracy at segment \( (-\Delta_c, \Delta_c) \), that is the determined integral – the primary function \( F_c(x) \), using the interpolation polynomial by Taylor or Maclaurin.

Given the fact that the primary function \( F_c(x) \) has three derivatives at segment \( (-\Delta_c, \Delta_c) \), we decompose this function into a Maclaurin series with a residual term in the Lagrange form:

\[
F_c(x) = \sum_{i=0}^{3} \frac{f^{(i)}(0)}{i!} (x-a)^i
\]

Considering (5):

\[
F^{(1)}(0) = f(0), \quad F^{(2)}(0) = f''(0), \quad F^{(3)}(x_2) = f'''(x_2).
\]

Thus, the absolute error of approximation \( \Delta_c \) is:

\[
\Delta_c = \frac{\Delta^3 f(x_c)(b-a)^3}{3n^3} = \frac{\Delta^3 f(x_c)(b-a)}{3n}.
\]

(7)

We obtain from (7) the number \( n \) of approximation sections (8) at curve \( f(x) \) of a neural or fuzzy model at section \( (a, b) \) with the assigned error \( \Delta_c \) approximated based on the formula of rectangles (9) or a trapezoidal formula (10):

\[
n = \Delta^2 f(x_c) \frac{b-a}{3\Delta_c},
\]

(8)

\[
F_c(x) = \frac{1}{n} \int_a^b f(x)dx = \frac{b-a}{n} \left[ f(x_0) + f(x_1) + \ldots + f(x_{2n-1}) \right] + \Delta_c.
\]

(9)

\[
\Delta_c = \frac{2}{n} \left[ f(x_0) + f(x_1) + \ldots + f(x_{2n-1}) \right] + \Delta_c.
\]

(10)

Similarly, given the absolute error \( \Delta \) along segment \( (-h, h) \) at \( a=0 \) and \( b=1 \), we compute area \( F(x) \) under curve \( f(x) \) by using a decomposition of the primary function \( F(x) \) into a Maclaurin series with a residual term in the Lagrange form:

\[
F(x) = \int_{-h}^{h} f(x)dx = F(h) - F(-h),
\]

(11)

where

\[
-h = x_{2k} - x_{2k-1}, \quad h = x_{2k+1} - x_{2k}.
\]

\[
\int_{-h}^{h} f(x)dx = F(h) - F(-h) = \left[ F(0) + f(0)h + f''(0) \frac{h^2}{2} + f'''(x_{2k}) \frac{h^3}{6} \right] - \left[ F(0) + f(0)h + f''(0) \frac{h^2}{2} - f'''(x_{2k}) \frac{h^3}{6} \right] = 2f(0)h + f^2(x_{2k}) \frac{h^3}{3} = f(0)2h + \frac{d^2 f(x_{2k})}{dx_{2k}^2} \frac{2(h)^3}{24},
\]

(12)

where \( x_{2k} \) is some point along segment \( (-h, h) \).

Since:

\[
d^2 f(x_{2k}) \frac{dx_{2k}^2}{3n^2} = \frac{\Delta^2 f(x_{2k})(h)^3}{3\Delta_{2k}^3},
\]

(13)

\[
\Delta_{2k} = \frac{1}{n},
\]

\[
\Delta_c = \frac{f(x_{2k})}{3n} = f(x_{2k}) + \Delta_c.
\]

(14)

then

\[
\int_{-h}^{h} f(x)dx = f(x_{2k}) - 2f(x_{2k-1}) + f(x_{2k-2}).
\]

(15)

where the absolute approximation error \( \Delta \) is equal to:

\[
\Delta = \frac{\Delta^2 f(x_{2k})h}{3n} = \Delta^2 f(x_{2k}) \frac{1}{3n}.
\]

(16)

We obtain from (16) the number \( n \) of approximation sections (17) at curve \( f(x) \) of a neural or fuzzy model \( \Delta_c \) at section \( (0, 1) \) with the assigned absolute approximation error \( \Delta_c \) using the formula of rectangles (18) or a trapezoidal formula (19):

\[
n = \frac{\Delta^2 f(x_{2k})}{3\Delta} = \frac{1}{h} = \frac{1}{x_{2k} - x_{2k-1}},
\]

(17)

\[
F(x) = \int_a^b f(x)dx = \left[ f(x_0) + f(x_1) + \ldots + f(x_{2n-1}) \right] + \Delta_c.
\]

(18)

\[
F(x) = \frac{1}{n} \left[ f(x_0) + f(x_1) + \ldots + f(x_{2n-1}) \right] + \Delta_c.
\]

(19)
Denote the relative error of approximation:
\[ \delta = \frac{3\Delta}{\Delta^2 f(x_3)} \]
\[ \frac{\Delta}{3}. \]

For example, for function \( y = \mu x^2 \), \( \Delta^2 f(x_3) = 0.02 \mu \) for \( n=10 \), \( h=0.1 \), and for any two adjacent sections:
\[ d^2 f(x_3) \left( \frac{\Delta}{h^2} \right) = 2\mu, \]
\[ \frac{\Delta}{h^2} = \delta = 0.02\mu = 0.000678\mu. \]

We obtain from (17):
\[ n = \frac{1}{h} = \frac{1}{x_{k+1} - x_{k-1}} = \frac{1}{\delta}. \]
\[ (21) \]

We obtain from expression (21) coordinates for the abscissa of modal values for the next point \( x_{k+1} \) in a model curve based on the coordinate for the preceding point \( x_{k-1} \) and the assigned value for error \( \delta \):
\[ x_2 = x_{k+1} \pm \delta. \]
\[ (22) \]

Table 1 gives, for \( x_{\text{max}} = 1, x_{\text{min}} = 0 \), the dependence of number \( n \) of the required sections of approximation and the number of neurons or terms \( m \) for fuzzy variables \( x, y \) depending on the preset values for approximation errors \( \delta \) or \( \Delta \).

<table>
<thead>
<tr>
<th>( \delta )</th>
<th>0.5</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
<th>0.02</th>
<th>0.01</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta )</td>
<td>0.00335\mu</td>
<td>0.00134\mu</td>
<td>0.00067\mu</td>
<td>0.000335\mu</td>
<td>0.000134\mu</td>
<td>0.000067\mu</td>
</tr>
<tr>
<td>( n )</td>
<td>2</td>
<td>5</td>
<td>10</td>
<td>20</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>( m )</td>
<td>3</td>
<td>6</td>
<td>11</td>
<td>21</td>
<td>51</td>
<td>101</td>
</tr>
</tbody>
</table>

4.2. Estimation of the number of neurons (terms) to ensure the accuracy of model implementation

At the next stage of research, we estimated the required number of neurons (terms), which provide for the necessary accuracy of model implementation based on the maximum deviation between the system and model curves at the section of approximation. The model of this system \( y \) will be searched for in the form of sections along straight lines that pass through points with coordinates \( x_1, y_1 \) and \( x_2, y_2 \), located along the approximated curve with the assigned maximum error of deviation \( \Delta_{\text{max}} \). Write down the equation of the straight line passing through two points of the curve with coordinates \( x_1, y_1 \) and \( x_2, y_2 \):
\[ \frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1} \]
\[ (23) \]

Considering (1), substitute in (12):
\[ \frac{y - f(x_1)}{f(x_1) - f(x_1)} = \frac{x - x_1}{x_2 - x_1} \]
\[ (24) \]

We obtain as a result:
\[ y = \frac{x - x_1}{x_2 - x_1} \left[ f(x_1) - f(x_1) \right] + f(x_1). \]
\[ (25) \]

Approximation error \( \Delta \) will be determined as the maximum deviation between the system \( f(x) \) and model \( f(x) \) curves along the approximation section from correlation (26):
\[ \Delta = y - y_1 = \frac{x - x_1}{x_2 - x_1} \left[ f(x_1) - f(x_1) \right] + f(x_1) - f(x). \]
\[ (26) \]

The maximum of error \( \Delta_{\text{max}} \) is determined from condition:
\[ \frac{\partial \Delta}{\partial x} = \frac{f(x_1) - f(x_1)}{x_2 - x_1} \frac{\partial f(x)}{\partial x} = 0, \]
\[ (27) \]

we obtain the dependence of coordinate of the next point \( x_2 \) based on the coordinate for the preceding point of curve \( x_1 \) and the assigned value for maximum error \( \Delta_{\text{max}} \).

An example of the implementation of the modeled system of guaranteed accuracy, described relative to the coordinate origin \( \theta (0, 0) \) by the equation of a non-central curve of second order in the form:
\[ a_1x^2 + 2a_2x + a_0 + 2b_0y = 0, \]
\[ (28) \]

was considered by authors in [5]. Transform of equation (28) is a curve (parabola), shifted relative to the coordinate origin:
\[ y = f(x) = \frac{a_2x^2}{a_0} - \frac{a_1}{2b_0a_0} + \frac{a_2}{2b_0}, \]
\[ (29) \]

The transformation of a parabola equation (28) and the transfer of coordinate origin to point \( O' \) with coordinates:
\[ x = x + \frac{a_1}{a_2}, \quad y = y + \frac{a_2}{2a_0}, \]
\[ (29) \]

produces the modeled system, described by the equation of parabola (29).
\[ y = \mu x^2, \]
\[ (29) \]

where \( \mu = \sqrt{1/2p} = -a_1/2b_0, p \) is the distance from the focus to the directrices of parabola.

The model of this system in [5] was found in the form of sections along straight lines that pass through points with coordinates \( x_1, y_1 \) and \( x_2, y_2 \) of parabola (29) with the assigned maximum error \( \Delta_{\text{max}} \) of deviation between curve (29) and straight line (25). Write down the equation of the straight line passing through two points with coordinates \( x_1, y_1 \) and \( x_2, y_2 \). Considering (29), substitute in (25):
\[ y = \mu x_1^2, \quad y_2 = \mu x_2^2. \]
The obtained result is:
\[
\frac{y - \mu x_i}{\mu} x_i = x_i - x_i + (x_i)(x_i) x_i = x_i - x_i + (x_i)(x_i) x_i
\]

hence:
\[
y - \mu x_i = \mu x_i + (x_i)(x_i) x_i - x_i x_i - x_i
\]

or
\[
y = \mu \left[ (x_i + x_i) x_i - x_i x_i - x_i \right]. \tag{30}
\]

Error of approximation \( \Delta \) is determined as a difference:
\[
\Delta = y - y_i = \mu \left[ (x_i + x_i) x_i - x_i x_i - x_i \right]. \tag{31}
\]

The maximum of error is determined from condition:
\[
\frac{\partial \Delta}{\partial x} \mu [x_i + x_i - 2x_i] = 0,
\]

hence:
\[
x = \frac{x_i + x_i}{2}. \tag{32}
\]

Substitute (32) into (31) and obtain the dependence of coordinates of the next point \( x_2 \) along the curve on the coordinate for the preceding point \( x_1 \) and the assigned value for maximum error \( \Delta_{\text{max}} \):
\[
\Delta_{\text{max}} = \mu \left[ \frac{(x_i + x_i)(x_i + x_i)}{2} - x_i x_i - (x_i + x_i)^2 \right].
\]

\[
4 \Delta_{\text{max}} = \mu \left[ (x_i + x_i)^2 - 4x_i x_i \right],
\]

\[
4 \Delta_{\text{max}} = \mu \left[ (x_i^2 + 2x_i x_i + x_i^2 - 4x_i x_i) = \mu \left[ (x_i^2 - 4x_i x_i + x_i^2) = \mu \left[ (x_i - x_i)^2 = \mu \right] \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \righ...
Fig. 3. A neural model of guaranteed accuracy

Fig. 4 shows the dependence charts of the modelled curve \( y(x, T_{x0}, T_{x1}, T_{x2}, T_{x3}) \) and the output variable \( y(T_{y0}, T_{y1}, T_{y2}, T_{y3}) \), whose modal values are located along the \( XY \) coordinate axes: \( a \) - evenly; \( b \) - unevenly.

Choose coordinates of the modal values for functions of the activation of neurons from condition (39), determining \( k_i \) from coefficients that are selected from \( k_i = y_i / x_i^2 \), and \( y_{ci} \) - from modal values for the modeled curve at approximation nodes \( x_i \) (Fig. 4).

In a given case, the nodal points of model curve 2 will be located along curve 1, which would provide for the assigned accuracy of modeling.

To test validity of the proposed solutions, we simulated different approximation algorithms in the environment Matlab/Simulink.

The results of simulation at approximation by fuzzy logic are shown in Fig. 5, by artificial neural networks - in Fig. 6.

The imitation simulation results indicate that the approximation error corresponds to values derived from analytical calculations.
6. Discussion of results of studying the analytical method for limiting the complexity of neural-fuzzy models of guaranteed accuracy

Therefore, we have proposed a solution on the choice of coordinates for the modal values of terms in fuzzy models, the number \( n \) of approximation sections and the number \( m \) of membership functions \( T_i \) of terms in a fuzzy model, the number of activation functions of neurons \( N_i \). Computation of the specified parameters is possible in line with formulae (21) to (22) and (33) to (39), depending on the assigned accuracy of model implementation.

In contrast to methods, examined in chapter 2, which imply the selection of values (settings) for parameters of neural-fuzzy models, the developed method solves this task analytically while assigning guaranteed accuracy. The advantage of the proposed method is the universality of its implementation in order to limit the complexity of fuzzy and neural models of systems, described by functions that have \( 2l \) derivatives.

The factors that remain unaddressed in this study include the uncertainty of the impact of noise and disturbances on the accuracy of a system curve approximation and their comparison to existing methods.

The further advancement of research could be accounting for the dynamics of work of neural-fuzzy models' elements over time.

8. Conclusions

1. We have devised a procedure for estimating the required number of neurons (terms) in a model that would provide for the necessary accuracy such that the area of a model curve approaches the system one. Evaluation of approximation error is performed based on the residual terms from decomposition of areas of the system function into a Maclaurin series.

2. We have devised a procedure for estimating the required number of neurons (terms), which provide for the required accuracy of model implementation based on the maximum deviation between the system and model curves at the approximation section. The derived dependences of error on the number of approximation sections and the number of terms for fuzzy variables make it possible to assign a predefined level of guaranteed accuracy for the implementation of models.

3. We have considered the solution to the applied task on constructing a fuzzy model and a model using ANN for a system with the assigned dependence \( y_x = \mu^2 \). The reported results from imitation simulation indicate exact match between the results and the estimated analytical values.

That testifies to the validity of the devised approach and the possibilities for its use in applied problems.

References


One of the basic problems in systems of object control in terms of input signal of their characteristics is classification problem. If the signal is represented as a time series, it is expedient to use neural networks to ensure high level of recognition accuracy. When a part of a signal or some of its values are lost at unknown time points, it is impossible