Development of neural network forecasting models of dynamic objects from observed data

Isamiddin Siddikov [1], Nodira Mamasodikova [2], Davronbek Khalmatov [3], Nargis Kadirova [3], Odiljon Mirjalilov [2], Go'zal Primova [4]

[1] *Tashkent State Technical University, 2, Universitetskaya street, Tashkent, 100095, Uzbekistan*
[2] *Fergana branch of Tashkent University of Information Technologies named after Mukhammad al-Khwarizmi, 185, Mustaqillik street, Fergana, 150118, Uzbekistan*
[3] *Tashkent Institute of Textile and Light Industry, 5, Shakhdjakhon street, Tashkent, 100100, Uzbekistan*
[4] *Samarkand branch of Tashkent University of information technologies named after Mukhammad al-Khwarizmi, 47A Sh. Mirzo Street, Samarkand, 140100, Uzbekistan*

Abstract

The paper investigates the issues of constructing predictive models of the state of dynamic models based on neural network structures, which are a powerful mathematical apparatus for approximating various types of functions and assessing the dynamics of changes in the states of the objects under consideration. A method is proposed for constructing predictive neural network models for the observed data of real-life objects using a hybrid application of the theory of nonlinear dynamics and nononical networks. The neural network predictive model built on the basis of the reconstructed phase trajectories of the process (attractor) will allow choosing the most informative characteristics, minimizing the architecture of the neural network and the dimension of the vector from the measured controlled signals. Based on the developed algorithms for preprocessing local areas of the reconstructed attractor and the selected machine learning method, a scheme for constructing a neural network predictive model of the dynamic behavior of a structurally complex system is proposed.

Keywords

Non-neural networks, forecasting, model, attractor, states, dynamic object, algorithm, time series, approximation

1. Introduction

One of the most important tasks in oil refining is the quality control of the resulting oil products. In real conditions, the operational control of oil refining processes, like most technological processes, is carried out using the results of the analysis of emerging production situations [1–6]. Analysis is usually carried out by collecting and primary processing of data collected from existing sensors, and intermediate laboratory analyzes.

However, the results of analyzes obtained in factory laboratories do not always have the required level of completeness and efficiency, they are rather rare and cannot be used for quality management in real time.

It should also be noted that in the majority of technical, including oil refining facilities, processes appear as a result of the interaction of various components, which significantly complicate the construction of adequate mathematical models based on a priori data. In such cases, it is more expedient

III International Workshop on Modeling, Information Processing and Computing (MIP: Computing-2021), May 28, 2021, Krasnoyarsk, Russia
EMAIL: isiddikov54@gmail.com (Isamiddin Siddikov); nodiramamasodikova@mail.ru (Nodira Mamasodikova); holdav@mail.ru (Davronbek Khalmatov); kadirova-nargis@gmail.com (Nargis Kadirova); odiljonmirjalilov@gmail.com (Odiljon Mirjalilov); primovaguzal@gmail.com (Go'zal Primova)
ORCID: 0000-0001-9681-5156 (Isamiddin Siddikov); 0000-0002-7605-879X (Nodira Mamasodikova); 0000-0002-2120-6257 (Davronbek Khalmatov); 0000-0002-8352-7232 (Nargis Kadirova); 0000-0003-4720-5532 (Odiljon Mirjalilov); 0000-0002-4200-4726 (Go'zal Primova)

to construct models of changes in the dynamics of the process under consideration, which are the observable parameters of a controlled complex system, giving high-quality predictive data of the modeled process [5–9]. Such models can be successfully applied in control problems, predicting the behavior of an object. In this regard, the development of a methodology for constructing predictive mathematical models of dynamic processes and phenomena is of paramount importance.

To solve the problem at the present time and are widely used, statistical methods have found. On the other hand, static methods are associated with the complexity of checking the fulfillment of the stochastic laws of the investigated implementation (time series) and the hypothesis of probabilistic characteristics, which can lead to inaccurate predictions.

2. Formation of the problem

Recently, the theory of nonlinear dynamical systems has been widely developed, based on the use of neural networks, which is a powerful mathematical apparatus for approximating a function, as well as for assessing the dynamics of changes in the state of the systems under study.

At the same time, the theory of nonlinear dynamics is not widely used in the construction of neural network models of processes. It should be noted that the analysis of the restored attractor (phase trajectories of the process) allows choosing the most informative characteristics, minimizing the structure of the neural network and the dimension of the vector - input signals.

The hybrid application of the methods of the theory of nonlinear dynamics and neural networks will make it possible to develop a methodology for constructing predictive models of actually functioning objects based on the observed data of attractors.

3. Method for solving the problem

To restore the attractor, you can use the time series obtained as a result of measuring the parameters of a really functioning structurally complex observable system, which is a function of the state of a dynamic system [3–11]:

$$x(t+1) = f(xlt)$$

$$y(t) = \phi(xlt)$$

Phase portrait restored as

$$z(t) = \{y(t), y(t-\tau), \ldots y(t-(m-1)\tau)\} = \{z,(1), z_n(t), \ldots z_m(t)\}$$

is an attractor of a dynamic system, providing a mapping of the state $z(t)$ to $x(t)$. Here $x(t)$ is the state vector of the original dynamical system. $y(t)$ - observed time series; $z_i(t)$ - is the ith component of the vector defining a point on the reconstructed attractor; τ - time of intersection of phase trajectories of the Poincaré section; m - dimensional reconstruction (model) $m \geq n^2 + 1$; n^2 – - Gaussian dimension [4 ÷ 12].

The operator model will be represented in the form:

$z = (t+1) = \Psi\left(z(\bullet)\right)$ - a vector of functions that maps the characteristics of the attractor in the area to the coordinate of the state vector in the next time. Here ψ is a transformation using the previous states to calculate the next one.

Parametric identification of a neural network model with a known (given) S one is carried out by searching for parameters W_s that allow a lot of error.

$$E = \sum_i \left(z(t+1)\right) - \psi_s\left(W_s, S, z(t)\right)$$

where ψ is the neural network estimate of the parameter of the dynamics of changes in the states of the system, S is the structure of the neural network model (the number of layers, the number of neurons in each layer, the type of the activation function); W_s - weight coefficients $z(t)$, - area on the restored attractor of the system.

To build a predictive model, we will investigate the model presented in the form $z(t+1)=\psi\left(\bar{z}\right)$ where $\bar{z}$ is a subset of points on the attractor of the system, ψ is a transformation operator designed to calculate the current state, applying the previous one.

We represent the neighborhood $\bar{z}$ in matrix form, where the central element is the current point of $z(t)$ the restored attractor of the system, and the central column is formed on the basis of ordering in ascending order of distance from the point $z(t)$ of its nearest neighbors [6–13]:

$$z_1(t_1), z_2(t_2), \ldots z_n(t_n)$$

$\bar{z}$ Matrix row elements are continuous path segments, and centers are the nearest neighbors of the point $z(t)$. The $\bar{z}$ matrix formed in this way has the form:

$$\bar{Z}=\begin{pmatrix} Z_{2r-1}(t_{2r-1}+k\Delta t) & \ldots & Z_{2r-1}(t_{2r-1}+\Delta t) & Z_{2r-1}(t_{2r-1}) & Z_{2r-1}(t_{2r-1}-\Delta t) & \ldots & Z_{2r-1}(t_{2r-1}-k\Delta t) \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ Z_1(t_1+k\Delta t) & \ldots & Z_1(t_1+\Delta t) & Z_1(t_1) & Z_1(t_1-\Delta t) & \ldots & Z_1(t_1-k\Delta t) \\ 0 & \ldots & 0 & Z(t) & Z(t-\Delta t) & \ldots & Z(t-k\Delta t) \\ Z_2(t_2+k\Delta t) & \ldots & Z_2(t_2+\Delta t) & Z_2(t_2) & Z_2(t_2-\Delta t) & \ldots & Z_2(t_2-k\Delta t) \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ Z_{2r}(t_{2r}+k\Delta t) & \ldots & Z_{2r}(t_{2r}+\Delta t) & Z_{2r}(t_{2r-1}) & Z_{2r}(t_{2r}-\Delta t) & \ldots & Z_{2r}(t_{2r}-k\Delta t) \end{pmatrix} \quad (1)$$

if *t*>-*t* then the elements of the $\bar{z}$ matrix are padded with zeros.

Zero elements in the central row of the $\bar{Z}$ matrix are explained by the absence of data at the points $Z(\bar{t})$ where $\bar{t}>t$.

For global models, the $\bar{Z}$ matrix is:

$$\bar{Z}=\begin{pmatrix} 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & Z(t) & Z(t-\Delta t) & \ldots & Z(t-k\Delta t) \\ 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \end{pmatrix}$$

For local models, the $\bar{Z}$ matrix can be written in the form (2) or (3)

$$\bar{Z}=\begin{pmatrix} 0 & \ldots & Z_{2r-1}(t_{2r-1}+\Delta t) & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & Z_1(t_1+\Delta t) & 0 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & \ldots & Z_2(t_2+\Delta t) & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & Z_{2r}(t_{2r}+\Delta t) & 0 & 0 & \ldots & 0 \end{pmatrix} \quad (2)$$

$$
\overline{Z} = \begin{pmatrix}
0 & \dots & Z_{2r-1}(t_{2r-1}+\Delta t) & Z_{2r-1}(t_{2r-1}) & 0 & \dots & 0 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots \\
0 & \dots & Z_1(t_1+\Delta t) & Z_1(t_1) & 0 & \dots & 0 \\
0 & \dots & 0 & Z(t) & 0 & \dots & 0 \\
0 & \dots & Z_2(t_2+\Delta t) & Z_2(t_2) & 0 & \dots & 0 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots \\
0 & \dots & Z_{2r}(t_{2r}+\Delta t) & Z_{2r}(t_{2r}) & 0 & \dots & 0
\end{pmatrix} \tag{3}
$$

Using matrix (2), local forecasting is carried out for the behavior of a nonlinear object based on averaging the values of the nearest neighboring points on the attractor of the system, while the matrix in (3) is used to determine the values of changes in these points. Mathematical models using attractors in the form of neighborhoods differing from (1), (2), (3) are synthetic [8 ÷ 11].

On the basis of the proposed methodology, an algorithm has been developed that made it possible to experimentally prove the effectiveness of the use of synthetic models in forecasting problems for the purpose of process control. The algorithm can be rearranged in six steps.

Step 1. Reconstruct the attractor of the system for a given time series.

Step 2. Form the next configuration of the $\overline{Z}$ matrix. Perform step 3 and step 4 N times:

Step 3. Build a neural network model.

Step 4. Get the forecast error on the test set for the constructed model.

Step 5. Evaluate the forecast quality for a given $\overline{Z}$ matrix configuration, taking into account the quality of N neural network models.

Step 6. Choose k $\overline{Z}$ matrix configurations for which the best results were obtained.

For the case under consideration, the structure of the neural network model has five layers (figure 1):

- layer 1 - inputs of the identification object;
- layer 2 - fuzzy terms used in the knowledge base;
- layer 3 - conjunction lines of the fuzzy knowledge base;
- layer 4 - rules grouped into classes d_j, $j = \overline{1, m}$;
- layer 5 - defuzzification operation, i.e. converting the results of fuzzy inference to a crisp number.

$$
y = \frac{y\mu^{d_1}(y) + y_1\mu^{d_2}(y) + \dots + y_{m-1}\mu^{d_m}(y)}{\mu^{d_1}(y) + \mu^{d_2}(y) + \dots + \mu^{d_m}(y)},
$$

where $\mu^{d_j}(y)$ is the function of membership of the output y to the class $d_j \in [y_{j-1}, y_j]$;

The number of nodes in a neural fuzzy network is determined as follows:

- layer 1 - by the number of inputs of the identification object;
- layer 2 - by the number of fuzzy terms in the knowledge base;
- layer 3 - by the number of conjunction lines in the knowledge base;
- layer 4 - by the number of classes into which the range of the output variable is divided.

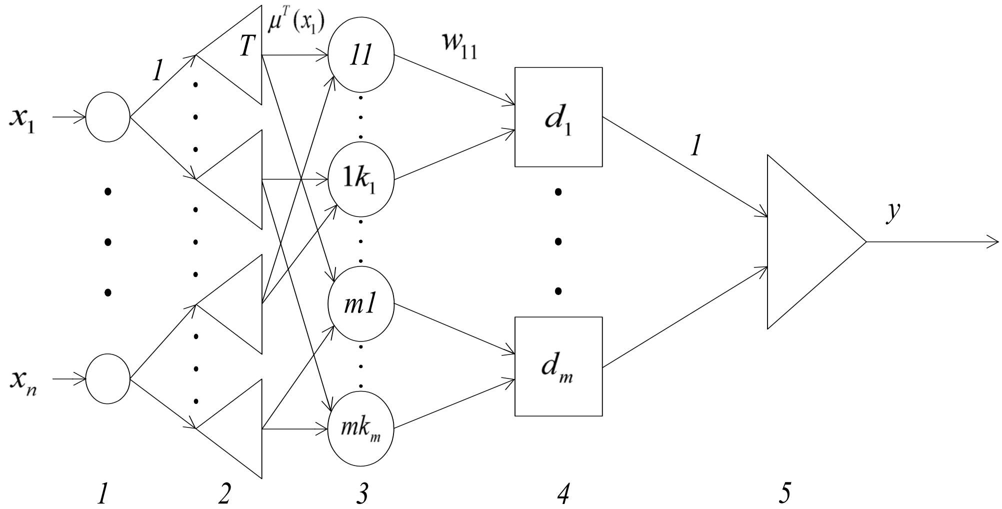

Figure 1: Neural fuzzy network structure

The graph arcs are weighted as follows:

- unit - arcs between the 1st and 2nd layers;
- functions of membership of the input to a fuzzy term - arcs between the 2nd and 3rd layers;
- weights of the rules - arcs between the 3rd and 4th layers;
- unit - arcs between the 4th and 5th layers;

The essence of training is to select such weights of arcs that minimize the difference between the results of neuro-fuzzy approximation and the real behavior of the object. The system of recurrent relations is used for training [7 - 14]:

$$w_{jp}(t+1) = w_{jp}(t) - \eta \frac{\partial E_t}{\partial w_{jp}(t)}, \tag{4}$$

$$c_i^{jp}(t+1) = c_i^{jp}(t) - \eta \frac{\partial E_t}{\partial c_i^{jp}(t)}, \tag{5}$$

$$b_i^{jp}(t+1) = b_i^{jp}(t) - \eta \frac{\partial E_t}{\partial b_i^{jp}(t)}, \quad j=\overline{1,m}, \ \ i=\overline{1,n}, \ \ p=k_j, \tag{6}$$

minimizing criteria

$$E_t = \frac{1}{2}(\hat{y}_t - y_t)^2 \tag{7}$$

used in the theory of neural networks, where: $\hat{y}_t$ and y_t - theoretical and experimental outputs of the object t at the m step of learning: $w_{jp}(t)$, $c_i^{jp}(t)$, $b_i^{jp}(t)$, - weights of the rules w and parameters of the membership (b/c) functions at t the m - step of learning; h - teaching parameter, which can be selected in accordance with the recommendations of work [10 ÷ 15].

The partial derivatives included in relations (4) - (7) characterize the sensitivity of the error (E_t) to a change in the parameters of a fuzzy ANN, and are calculated as follows: ε

$$\frac{\partial E_t}{\partial w_{jp}} = \varepsilon_1 \varepsilon_2 \varepsilon_3 \frac{\partial \mu^{d_j}(y)}{\partial w_{jp}}, \tag{8}$$

$$\frac{\partial E_t}{\partial w_i^{jp}} = \varepsilon_1 \varepsilon_2 \varepsilon_3 \varepsilon_4 \frac{\partial \mu^{jp}(x_i)}{\partial c_i^{jp}}, \tag{9}$$

$$\frac{\partial E_t}{\partial b_i^{jp}} = \varepsilon_1 \varepsilon_2 \varepsilon_3 \varepsilon_4 \frac{\partial \mu^{jp}(x_i)}{\partial b_i^{jp}}, \tag{10}$$

where

$$\varepsilon_1 = \frac{\partial E_t}{\partial y} = y_t - \widehat{y}_t, \sqrt{2} \tag{11}$$

$$\varepsilon_2 = \frac{\partial y}{\partial \mu^{d_j}(y)} = \frac{\overline{\partial}_j \sum_{j=1}^{m} \mu^{d_j}(y) - \sum_{j=1}^{m} \overline{\partial}_j \mu^{d_j}(y)}{\left(\sum_{j=1}^{m} \mu^{d_j}(y) \right)^2}, \tag{12}$$

$$\varepsilon_3 = \frac{\partial \mu^{d_j}(y)}{\partial \left(\prod_{i=1}^{n} \mu^{jp}(x_i) \right)} = w_{jp}, \tag{13}$$

$$\varepsilon_4 = \frac{\partial \left(\prod_{i=1}^{n} \mu^{jp}(x_i) \right)}{\partial \mu^{jp}(x_i)} = \frac{1}{\mu^{jp}(x_i)} \prod_{i=n}^{n} \mu^{jp}(x_i), \tag{14}$$

$$\frac{\partial \mu^{d_j}(y)}{\partial w_{jp}} = \prod_{i=1}^{n} \mu^{jp}(x_i), \tag{15}$$

$$\frac{\partial \mu^{jp}(x_i)}{\partial c_i^{jp}} = \frac{2c_i^{jp}\left(x_i - b_i^{jp}\right)^2}{\left(\left(c_i^{jp}\right)^2 + \left(x_i - b_i^{jp}\right)^2 \right)^2}, \tag{16}$$

$$\frac{\partial \mu^{jp}(x_i)}{\partial b_i^{jp}} = \frac{2\left(c_i^{jp}\right)^2\left(x_i - b_i^{jp}\right)}{\left(\left(c_i^{jp}\right)^2 + \left(x_i - b_i^{jp}\right)^2 \right)^2} \tag{17}$$

Similar to the rule, the fuzzy ANN learning algorithm consists of two phases. In the first phase, the model value of the object's output y, is calculated corresponding to the given network architecture. In the second phase, the residual value E_t is calculated and the weights of interneuronal connections are recalculated using formulas (8) - (17) [6 ÷ 10].

As you can see, the use of fuzzy ANN can significantly reduce the time spent in solving the problem of identifying nonlinear objects. In addition, it should be noted that learning the proposed fuzzy ANN allows you to switch to a new way of processing experimental information: - obtaining a fuzzy rule base. The principal of this method is the convenience of interpreting the results obtained.

To assess the characteristics of the local neighborhood, a ψ, transformation technique is proposed, which is presented as a sequential execution of the stages of preprocessing and calculating the output characteristics of the existing model:

$$\psi\left(\overline{Z}\right) = \psi_n\left(\psi_p\left(\overline{Z}\right)\right)$$

where ψ_p is the local neighborhood preprocessing function; ψ_n - neural network model. It was found that preprocessing ψ_p techniques solve the following tasks:

Filtering, accounting for the forecast of alternative models for the current situation (position on the attractor) and accounting for the forecast error of identical in structure for similar situations. The main idea of the proposed combined preprocessing technique is to assess the characteristics of a local neighborhood aimed at solving each of the selected problems.

4. Conclusion

The complex of theoretical and practical results obtained in the work made it possible to create methods for constructing predictive models based on the analysis of attractors of nonlinear dynamic systems, reconstructed on the basis of experimental data. Wherein:

- developed a dynamic model, identified by a one-dimensional implementation based on the analysis of local neighborhoods of the reconstructed attractor, providing the construction of a local, global and synthetic forecast;
- developed a scheme for constructing a neural network predictive model of the dynamic behavior of a structurally complex system based on the developed algorithms for preprocessing local areas of the reconstructed attractor and the selected machine learning method.

5. References

[1] R. A. Aliev, R. R. Aliev, The theory of intelligent systems, Baku, 2001.

[2] A. F. Egorov, T. V. Savitskaya, Methods and models for risk analysis and chemical production safety management, J. Theoretical Foundations of Chemical Technology 44 (2010) 341-353.

[3] D. A. Pospelov, Fuzzy sets in models of control and artificial intelligence, Moscow, 1986.

[4] S. A. Orlovsky, Decision-making problems with fuzzy initial information, Moscow, 1981.

[5] A. V. Andreychikov, O. N. Andreychikova, Intelligent information systems, Moscow, 2004.

[6] N. G. Yarushkina, Foundations of the theory of fuzzy and hybrid systems, Moscow, 2004.

[7] I. Kh. Siddikov, N. Yu. Mamasodikova, G. R. Alimova, Automation System of the Technologikal Monitoring of Oil Enterprises Based on a Modeling System, Journal of Advanced Research in Science, Engineering and Technology 6 (2019) 11891-11896.

[8] I. X. Siddikov, D. M. Umurzakova, Mathematical Modeling of Transient Processes of the Automatic Control System of Water Level in the Steam Generator, Journal of Mechanical Engineering 7 (2019) 139-46. DOI: 10.13189 / ujme.2019.070401.

[9] I. X. Siddikov, D. M. Umurzakova, Features of automatic control of technological parameters of water level in the drum steam boilers, Journal of Jiaotong University 54 (2019) 1-10. DOI: 10.35741 / issn.0258-2724.54. 3.1.

[10] I. X. Siddikov, N. Yu. Mamasodikova, M. M. Khalilov, A. K. Amonov, G. B. Sherboboyeva, Formalization of the task of monitoring the technological safety of industrial facilities in conditions of indistinctness of the initial information, Journal of Physics: Conference Series 1679 (2020) 032022. doi:10.1088/1742-6596/1679/3/032022.

[11] Kh. O. Urinov, A. K. Amonov, Kh. A. Jumanov, I. Sh. Mujdibaev, N. Yu. Mamasodikova, Abnormal behavior of the torque in the system of magnetitemaghemite $\gamma - Fe_2O_3 - Fe_3O_4$,, Journal Materials Science and Engineering 919 (2020) 022012. doi:10.1088/1757-899X/919/2/022012.

[12] I. H. Sidikov. M. O. Atajonov, S. T. Yunusova, G. M. Nashvandova, Adaptive analytical control of technological parameters based on the probability method of oil refining installations, European Science Review, Scientific journal (2020) 1–2.

[13] I. H. Sidikov, M. O. Atajonov, S. B. Atajonova, Fuzzy-Situational Diagnostics of Technological Safety of Petrochemical Plants, J. EAST. 4 (2019) 182-186. doi: 10.33564/ IJEAST.2019.v04i07.030.

[14] S. T. Yunusova, D. A. Halmatov, M. O. Atajonov, U. O. Huzanazarov, Formalization of the Cotton Drying Process Based on Heat and Mass Transfer Equations, IIUM Engineering Journal 21 (2020) 256–265. doi.org/10.31436/iiumej.v21i2.1456.

[15] I. H. Sidikov, M. O. Atajonov, S. T. Yunusova, G. M. Nashvandova, Adaptive analytical control of technological parameters based on the probability method of oil refining installations, European Science Review Scientific journal 1 (2020) 78-83. doi.org/10.29013/ESR-20-1.2-78-83.