

## Research Article

# The Advanced Algorithmic Method for Navigation System Correction of Spacecraft

Danhe Chen <sup>1</sup>, K. A. Neusypin,<sup>2</sup> Xiang Zhang,<sup>1</sup> and Chuangge Wang<sup>1</sup>

<sup>1</sup>Nanjing University of Science and Technology, 210094, China

<sup>2</sup>Bauman Moscow State Technical University, 105005, Russia

Correspondence should be addressed to Danhe Chen; juliachen@njust.edu.cn

Received 29 April 2019; Accepted 25 June 2019; Published 15 July 2019

Academic Editor: Akhil Garg

Copyright © 2019 Danhe Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper an advanced method for the navigation system correction of a spacecraft using an error prediction model of the system is proposed. Measuring complexes have been applied to determine the parameters of a spacecraft and the processing of signals from multiple measurement systems is carried out. Under the condition of interference in flight, when the signals of external system (such as GPS) disappear, the correction of navigation system in autonomous mode is considered to be performed using an error prediction model. A modified Volterra neural network based on the self-organization algorithm is proposed in order to build the prediction model, and the modification of algorithm indicates speeding up the neural network. Also, three approaches for accelerating the neural network have been developed; two examples of the sequential and parallel implementation speed of the system are presented by using the improved algorithm. In addition, simulation for a returning spacecraft to atmosphere is performed to verify the effectiveness of the proposed algorithm for correction of navigation system.

## 1. Introduction

The autonomous navigation system of spacecraft can be used to control the spacecraft without relying on ground-based support and to determine the position, speed, and altitude of spacecraft in real time by measurement equipment aboard. The navigation system of spacecraft as a core for space engineering mainly provides information in the stages of its orbit entry, reentry, orbit change, and large altitude maneuvers, significantly depending on the data processing ability of the system algorithm [1, 2]. However, under some certain circumstances with interferences out of measurement systems, the correction of navigation system may not function in autonomous mode. And it is not possible to make the state prediction of maneuvering object using a priori mathematical models. In order to improve the performance and reliability of equipment in satellite in flight, various approaches have been studied including the development of algorithmic method for the correction of navigation system. Generally, the satellite navigation system is composed of inertial navigation system (INS) and global positioning system (GPS) receiver. When such a system operates, signals

from the space radio navigation system GPS may lose in flight due to the effect of active and passive interference [3, 4]. It is difficult to use general mathematical approximation functions to describe and predict the state of spacecraft. In this case, compacted model of error prediction algorithms which can reduce the computational costs is required to develop for spacecraft in autonomous mode [5].

Neural networks consist of a large number of interconnected processing elements which are called neurons, operating as microprocessors [6, 7]. Recently, several new methods have been developed with the concept of neural networks including prediction model [8–11]. For example, an identification method for nonlinear dynamic systems was proposed [12]. The feedforward neural network with the structure of Volterra system possesses more adjustable parameter than the original system to enhance the modeling capacity [13, 14]. Bukharov O. E proposed a development of decision support system based on neural networks and a genetic algorithm. They have justified the use of general-purpose computing on graphics processing units (GPGPU) for decision support system [15] and developed a general formulation of the prediction and estimation problems for

a class of weakly structured problems using interval neural networks and genetic algorithms. He showed two examples of application of the developed system for solving urgent problems [16]. In [17] a new structure of innovative decision support systems (DSS) with advantages of using neural networks to provide users precise prediction and optimal decisions has been developed. The applying interval neural networks for calculations with interval data makes it possible to use their DSS in a wide range of complicated tasks.

Actually, the neural network has been applied to the adaptive control of aircraft in recent years. WANG Qing et al. proposed an antiwindup adaptive control method of aircraft based on neural network and pseudocontrol hedging, according to the unfit flight control of conventional adaptive control on an actuator with magnitude saturation and rate saturation [18]. LIN Jian et al. studied a model reference adaptive control based on improved BP neural networks together with dynamic inversion and neural networks, which increases the efficiency of the adaptive algorithm and achieves the anti-interference purpose [19]. In [20] a model of reference adaptive control based on BP network that transfer function can optimize by itself has been put forward. Recent research results show that neural networks are very effective for modeling the complex nonlinear systems, especially those which are hard to be described in mathematical form [21, 22]. However, when neural network is used to process a complex system, it always takes a long time due to the requirement of a large number of neurons to process [23]. In order to reduce operation time of the neural network, it suggested using the self-organization algorithm and parallel network algorithm [24]. In this paper advanced algorithmic techniques are proposed for correcting the autonomous navigation system of spacecraft combined with neural networks; methods are presented instead of traditional algorithm for implementation on-board of the dynamic object. A modified Volterra network structure is newly proposed specially in model building of spacecraft, which could significantly accelerate the processing efficiency of neural network and increase the navigation precision.

The structure of this paper is presented as follows. An algorithm of building a prediction model of compensation for autonomous INS errors is developed in Section 2. A method of amplitude-frequency search based on the basic function is given in Section 3. Section 4 is concerned with the modification of the Volterra neural network using the method of self-organization, and a modified Volterra network is presented. The last section discusses the computer simulation results considering flight of a return spacecraft and the finally conclusions are given.

## 2. Algorithm of Building a Prediction Model

**2.1. Selection of the Reference Function.** Generally, different approaches to the prediction differ in terms of the amount of a priori information necessary for the prediction about the object under study. If considering an autonomous INS functioning for a long period (more than 6 hours), it is not available to correct INS from external devices and systems.

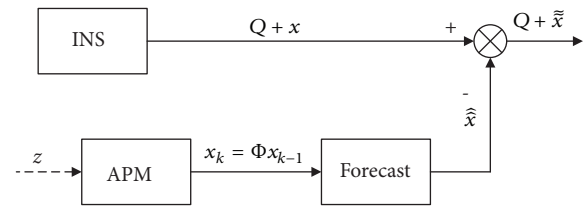


FIGURE 1: Structure diagram of the INS with the application of APM when external sensors are disconnected.

The main task to compensate for the errors of autonomous INS using only internal information herein is proposed. It also assumed that the autonomous operation mode of INS proceeded the system operation period in the correction mode of satellite system. Structure diagram of INS considering the algorithm of building a prediction model (APM) when external sensors are disconnected is shown in Figure 1.

In addition, dynamic objects usually could move in space by different trajectories to effectively perform the tasks. In process of designing the control systems for dynamic objects operating in an actively counteracting environment, as a rule, it is possible not only to perform various maneuvers, but also to control the basis of prediction of the object state.

In practical applications, predicting the state of the maneuvering object by using a priori mathematical models is not possible and reliable. When a dynamic object functions under stochastic conditions, the amount of a priori information about the object is usually minimal. Therefore, it is advisable to use the self-organization approach for extrapolation.

Self-organization algorithm allows building a mathematical model without a priori indication of the rules of the object. The developer of the mathematical model should set the ensemble of selection criteria (self-organization criteria) of the model selection, then the mathematical model of the optimal complexity is selected automatically. Furthermore, implementation of the self-organization algorithm is assumed on board of the dynamic object. Typically, such algorithms are presented fairly strict requirements for speed, compactness, and ease of implementation in a computer. These requirements are especially important when predicting the state of highly maneuverable dynamic objects.

The principle of self-organization algorithm for models is formulated as follows: with a gradual increase in the complexity of models, the value of internal criteria (in the presence of noise) decreases monotonically. Under the same conditions, all values of external criteria pass through their minima (extremums), making it possible to determine the model with optimal complexity, which is unique for each external criterion.

For self-organization method, the following three conditions must be met:

- (1) An initial organization (a set of support functions).
- (2) A mechanism for random changes (mutations) of this organization (a set of models-applicants).

- (3) A selection mechanism by which these mutations can be evaluated in terms of their usefulness for improving the organization (self-organization algorithm).

To a large extent, success of the self-organization modeling depends on the choice of reference functions class. If the reference functions such as the structure of object cannot be restored using a combination of particular models, then the approximation problem is still solved; however, the result is often suitable only for prediction, and not for object identification, since it is not a physical model of the object. But, the task of selecting a description is solvable if the class of reference functions is chosen sufficiently general. The available a priori information allows us to restrict a few types of reference functions and the model structures derived from them.

In the self-organization model, such reference functions as power polynomials, trigonometric functions, and exponential functions can be applied. If several types at the same time are included in the system of reference functions, then mixed functions containing the sum or product of power polynomials and exponential functions can be obtained.

2.2. *The Selection Criteria for the Model.* According to the principle of Gödel external complement, it is necessary to choose a criterion for the selection of model with optimal complexity. In order to solve the problem, we need to divide the data table into two parts A and B. Part A is a training sample and part B is a test sample. Sometimes the table is also divided into the third part C, an examination sample, which is used to evaluate various models, and it may also serve to select the optimal division into a training and verification sequence. With such a partition, optimal models are selected from a set of functions based on the training sequence, and one or two better functions will be selected during the criterion test.

The following criteria are most commonly used.

(1) *The Criterion of Minimum Displacement: Consistency.* According to this criterion, evaluation of model is formulated based on the data observed at a certain interval or at a certain observation point; it should coincide as close as possible with the model obtained from another observation interval or at another observation point.

One of the criteria is as follows:

$$n_{ob}^2 = \frac{1}{n} \sum_{t \in N} (y_t^A - y_t^B)^2 \rightarrow \min \quad (1)$$

(2) *The Criterion of Regularity.* The model of standard deviation on the test sample is defined as

$$\Delta^2(B) = \frac{\sum_{t \in N} (y_t^M - y_t)^2}{\sum_{t \in N} y_t^2} \rightarrow \min \quad (2)$$

If we assume that under a constant complex of conditions, a good approximation in the past guarantees a good enough approximation in the near future, then the regularity criterion can be especially recommended for a short-term prediction, since the solution obtained on the new implementation method only produces a small deviation, and thus the

established model will be regular, i.e., insensitive to small changes in the initial data. In this case, important variables may be lost during the selection process, the influence of which will be indirectly taken into account through other variables.

(3) *Balance Criteria.* With a set of constant conditions and in the absence of disturbances in structure of the object, the laws (relations of characteristic variables) acting at the observed time interval remain in the future. According to this criterion, one model is chosen from all the ones obtained at a certain time interval, and it is best corresponding to the given regularity. Let  $f(u_1(t), u_2(t), \dots, u_s(t)) = 0$  be balance functions (the associated variables  $u_i(t)$ ,  $i = 1 : s$ ). From the set of all prediction models for variables  $u_i(t)$ , the model should be chosen for which this ratio is best performed in the extrapolation interval. The imbalance of variables can be defined as  $b_i = f(u_1(t_i), \dots, u_s(t_i))$ , where  $t_i$ -moment at the prediction interval. The balance criterion allows choosing the best prediction from possible trends for each predictable process. In many cases, a function that represents the relationship between variables is easy to learn from physical representations. In other cases, the relationship of variables can be determined by using group argument algorithms.

(4) *The Criterion of Simplification.* As a model of optimal complexity, model with a smaller number of arguments is chosen with a simple reference function. Simplification of the self-organization algorithm can be carried out by reducing the number of basis functions, cutting down the selection by including in the ensemble of selection criteria, which is composed of any criterion for the simplicity of model.

When the method of self-organization is used, the predictive model can be written as

$$\Phi(x) = \sum_{i=1}^N a_i \mu_{ni}(f_i x) \quad (3)$$

where  $n$  is the number of basis functions in the model;  $\mu_n$  are basis functions from the parametrized set  $F_p$ ,  $F_p = \{a_i \mu_i(f_i x) \mid i = 1, L\}$ , a set of basis functions. Each basis function is associated with a two-dimensional vector of parameters  $(a, f)^T$ , where  $a$  is amplitude;  $f$  is frequency.

As a model of optimal complexity, the one with smaller number of arguments is chosen with a simple reference function.

$$\Phi(x, a) = \sum_{i=1}^N a_i \mu_i(x) \quad (4)$$

where  $N$ -number of basic functions,  $a_i \neq 0$ .

The criterion is defined as

$$\tau(\Phi) = \sum_{i=1}^N \delta(a_i), \quad \tau \rightarrow \min \quad (5)$$

where  $\tau$ -criterion function.

$$\delta(a_i) = \begin{cases} 1, & a_i > \varepsilon \\ 0, & a_i \leq \varepsilon \end{cases} \quad (6)$$

where the  $\varepsilon$  tends to be infinitesimal (chosen according to the actual situation).

The criterion of model simplification helps to significantly simplify the implementation of self-organization algorithm in the special on-board calculator of spacecraft. To reduce computational costs and obtain compact models in the self-organization algorithm, an original criterion for model simplification is included in the ensemble of selection criteria, which tends to be a more compact model with similar values from the ensemble of selection criteria. Using the constructed nonlinear model, the state of the object (INS errors) is predicted in the autonomous mode, i.e., in the absence of measurements from external sources.

To predict the state of the object under study, a mathematical model that contains all the necessary information about its parameters should be formed, and its state changes during a given period of time. In particular, if we take a sensor reading at certain (not necessarily equal) intervals, then the measurement results can be written down as  $\Omega = \{(x_i, y_i) \mid i = 1, \dots, n\}$ . The information presented in this way on the change of one of the object parameters is represented a sample (further assumes that,  $x_i < x_j$  when  $i < j$ , and  $x$  is considered as certain analog time).

The essence of forecasting is to build a model (or select from a set) that best meets the specified criteria and further calculate its values at the points  $x > x_n$ . Process of building such a model can be formally divided into separated stages: the first stage is to define the parameterized class of models, in which the search is performed. Examples include methods for finding one of functions belonging to a selected set and depending on a certain parameter vector or the method of sequential identification described below. Methods based on building impulse reactions (weight functions) are also widely used, and most of these methods widely apply the theory of statistics and random process.

**2.3. Identification of the Basis Function.** Here we introduce the criterion for identifying basis models:

$$I_i(a, f) = \sum_{k=1}^n (y_k - a\mu_i(fx_k))^2 \quad (7)$$

where  $\mu_i$  is a basis function of  $F_p$  and  $(y_k, x_k) \in \Omega$ .

The identification of 1-st basis model is the process of minimization of the criterion for identifying the 1-st basis function in frequency and amplitude:  $\min_{a,f} I_i(a, f)$ . Due to the impossibility of direct application of gradient methods (in most cases, functions have a large number of local minima), they must be used in combination with the Monte Carlo method [18], which greatly slows down the process. In order to avoid the two-dimensional minimization, we have limited the standard deviation. The expression for  $I_i(a, f)$  is a quadratic form with respect to the amplitude:

$$\begin{aligned} I_i(a, f) &= \sum_{k=1}^n (y_k - a\mu_i(fx_k))^2 \\ &= \sum_{k=1}^n (y_k^2 - 2y_k a\mu_i(fx_k) + a^2 \mu_i^2(fx_k)) \end{aligned} \quad (8)$$

The square of the difference under the sum is differentiated by  $a$ . At this point  $(I_i)'_a = 0$ , it can be written down as follows:

$$\sum_{k=1}^n (-2y_k\mu_i(fx_k) + 2a\mu_i^2(fx_k)) = 0 \quad (9)$$

And from the liner relationship with respect to  $a$ , we have

$$a = \frac{\sum_{k=1}^n y_k\mu_i(fx_k)}{\sum_{k=1}^n \mu_i^2(fx_k)} \quad (10)$$

Equation (10) can be used to reduce the dimension of the minimization region by the Monte Carlo method. The tests performed showed that using (10) always found a deeper minimum even with fewer points than with two-dimensional minimization.

After the random search, an assumption is made that the point found especially in a unimodal vicinity of the global minimum, after which a clarification will be obtained by gradient method. When solving the problem of identification, a model of the form  $a \cdot g(f \cdot x)$  is obtained. This could be used to build one of the simplest method of approximation, sequential identification. In the model building, all basic functions are identified firstly, as the best one from criterion, then, the difference (first remainder) of sample and model values is calculated; later, the remainder is passed through the identification algorithm again, the second remainder is found, etc. This process should be continued until the criterion value decreases.

It is also noticed that if the minimum value of criterion had a linear function in the first step, then the further process will no longer be able to change the overall process, while a completely different situation is observed when applying selective algorithms; for example, the curve obtained for the same sample, using the proposed method, does not have a dominant linear trend. Self-organization algorithms are multirow algorithms based on the selection hypothesis, which states that models do not pass the threshold of self-selection (if the corresponding criterion is chosen optimally) and do not get in the formation of best models in the next row.

Assume that the first selection series consist of  $N$  basis functions, each of which associates one parameter, its amplitude. Power functions are used in order to obtain a polynomial at the output of algorithm, and trigonometric functions are the Fourier series.

In each new row models are built as a linear combination of two pair different models from the previous row and constant. Thus, the combinations of the following type are formed from  $N$  models  $C_N^2$ :

$$y_i = a_{0,j}1 + a_{1,j}g_{0,j}(x_i) + a_{2,j}g_{1,j}(x_i), \quad j = 1 \dots C_N^2 \quad (11)$$

If sum up all these  $N$  equations with respect to  $i$ , we can obtain the system of so-called normalized Gauss equations:

$$\sum_{i=1}^N y_i = a_0 N + a_1 \sum_{i=1}^N g_{0,j}(x_i) + a_2 \sum_{i=1}^N g_{1,j}(x_i)$$

$$\begin{aligned} \sum_{i=1}^N y_i g_{0,j}(x_i) &= a_0 \sum_{i=1}^N g_{0,j}(x_i) + a_1 \sum_{i=1}^N g_{0,j}^2(x_i) \\ &+ a_2 \sum_{i=1}^N g_{0,j}(x_i) g_{1,j}(x_i) \end{aligned} \quad (12)$$

$$\begin{aligned} \sum_{i=1}^N y_i g_{1,j}(x_i) &= a_0 \sum_{i=1}^N g_{1,j}(x_i) + a_1 \sum_{i=1}^N g_{0,j}(x_i) g_{1,j}(x_i) \\ &+ a_2 \sum_{i=1}^N g_{1,j}^2(x_i) \end{aligned}$$

Since linear combinations of models are considered, we need a free term in each equation due to the fact that it is better to approximate on a given segment by a plane rather than a subspace constructed as a linear extension of the set of basis functions. If a constant is introduced in the basis, then the free term in the equations can be discarded. Thus, we have to search for two parameters for each pair of models.

$$y_i = a_{0,j} g_{0,j}(x_i) + a_{1,j} g_{1,j}(x_i), \quad j = 1 \dots C_N^2$$

$$\sum_{i=1}^N y_i g_{0,j}(x_i) = a_0 \sum_{i=1}^N g_{0,j}^2(x_i) + a_1 \sum_{i=1}^N g_{0,j}(x_i) g_{1,j}(x_i) \quad (13)$$

$$\sum_{i=1}^N y_i g_{1,j}(x_i) = a_0 \sum_{i=1}^N g_{0,j}(x_i) g_{1,j}(x_i) + a_1 \sum_{i=1}^N g_{1,j}^2(x_i)$$

Calculations are stopped when the minimum of the ensemble of criteria is reached, and result is the best model in the last row. Assume that there is a sample of  $N$  points; divide it into two parts:  $A$  is the training part on which the models are built and  $B$  is the verification sequence. The mean square error calculated for the sequence  $B$  is not concluded in the model building by the criterion of regularity.

$$\Delta^2(B) = \frac{\sum_{i=1}^{N_B} (y_i - q_i)^2}{\sum_{i=1}^{N_B} y_i^2} \quad (14)$$

Here,  $y_i$  is the sample values, and  $q_i$  is the model values computed at the point  $x_i$ . A description of the methods for dividing the original sample can be found. Suppose that  $\alpha$  is the extrapolation coefficient,  $A$  and  $B$  are two parts of the input sequence, and the value is described by (15):

$$n_b^2 = \frac{\sum_{i=1}^{\alpha N} (q_A - q_B)^2}{\alpha \sum_{i=1}^{N_B} y_i^2} \quad (15)$$

Accordingly, the criterion of minimum displacement is described above, which helps selecting the least sensitive to change in the input sample of the model. It allows solving the problem of restoring the law using noisy data. The convergence criterion of step-by-step integration of finite-difference models is as follows:

$$I = \frac{\sum_{i=1}^N (q_i - y_i)^2}{\sum_{i=1}^N y_i^2} \quad (16)$$

$I$  is a step-by-step integration error in the interpolation interval.

In practice, usually none of the above criteria is used, but instead constitutes a so-called ensemble of criteria. In many problems, ensembles of the following type have proved themselves well:

$$\begin{aligned} &\sqrt{I^2 + \Delta^2} \\ &\sqrt{\Delta^2 + n_b^2} \end{aligned} \quad (17)$$

The greatest freedom of action provides the ensemble of the form as

$$w_I I + w_\Delta \Delta^2 + w_{n_b} n_b^2 \quad (18)$$

where  $w_\alpha$  is the weight of the relevant criteria.

The application of this type of criteria selection allows changing the weights of its individual components during the operation of algorithm and performing corrections in the process of work by levels.

### 3. Description of the Amplitude-Frequency Search Method

The first step of the algorithm consists in identifying the basis functions by the corresponding criterion (see (7)). Further, based on the above discussed, each next level will be composed of combinations of models by the previous level as follows:

$$a_{n_0} \mu_{n_0}(f_{n_0} x) + a_{n_1} \mu_{n_1}(f_{n_1} x) \quad (19)$$

where  $n_0, n_1$  are the model numbers from the previous level,  $f_n$  is frequency,  $a$  represents amplitude, and  $\mu$  is the model from the previous level.

It is obvious that there are four variables for each pair model; thus, we need to solve the following problem in order to find the final model:

$$\min_{a_0, f_0, a_1, f_1} K(a_{0,j} g_{0,j}(f_{0,j} x_i) + a_{1,j} g_{1,j}(f_{0,j} x_i)) \quad (20)$$

where  $K$  is the value of the selection criterion. It is rather difficult to carry out minimization by Monte Carlo method in four-dimensional space. However, we could write down the analog of Gaussian system as follows:

$$\begin{aligned} \sum_{i=1}^N y_i \mu_{n_0}(f_{n_0} x_i) &= a_0 \sum_{i=1}^N \mu_{n_0}^2(f_{n_0} x_i) \\ &+ a_1 \sum_{i=1}^N \mu_{n_0}(f_{n_0} x_i) \mu_{n_1}(f_{n_1} x_i) \end{aligned} \quad (21)$$

$$\begin{aligned} \sum_{i=1}^N y_i \mu_{n_1}(f_{n_1} x_i) &= a_0 \sum_{i=1}^N \mu_{n_0}(f_{n_0} x_i) \mu_{n_1}(f_{n_1} x_i) \\ &+ a_1 \sum_{i=1}^N \mu_{n_1}^2(f_{n_1} x_i) \end{aligned}$$

Now this system can be solved with respect to  $a_n$ , since they are linearly in (21):

$$a_0 = \frac{\sum_{i=1}^N \mu_{n_1}^2(f_{n_1} x_i) \sum_{i=1}^N y_i \mu_{n_0}(f_{n_0} x_i) - \sum_{i=1}^N \mu_{n_0}(f_{n_0} x_i) \mu_{n_1}(x_i f_{n_1}) \sum_{i=1}^N y_i \mu_{n_1}(f_{n_1} x_i)}{\sum_{i=1}^N \mu_{n_0}^2(f_{n_0} x_i) \sum_{i=1}^N \mu_{n_1}^2(f_{n_1} x_i) - \left(\sum_{i=1}^N \mu_{n_0}(f_{n_0} x_i) \mu_{n_1}(x_i f_{n_1})\right)^2} \quad (22)$$

$$a_1 = \frac{\sum_{i=1}^N \mu_{n_0}^2(f_{n_0} x_i) \sum_{i=1}^N y_i \mu_{n_1}(f_{n_1} x_i) - \sum_{i=1}^N \mu_{n_0}(f_{n_0} x_i) \mu_{n_1}(x_i f_{n_1}) \sum_{i=1}^N y_i \mu_{n_0}(f_{n_0} x_i)}{\sum_{i=1}^N \mu_{n_0}^2(f_{n_0} x_i) \sum_{i=1}^N \mu_{n_1}^2(f_{n_1} x_i) - \left(\sum_{i=1}^N \mu_{n_0}(f_{n_0} x_i) \mu_{n_1}(x_i f_{n_1})\right)^2}$$

Based on these equations described above, we can minimize in a two-dimensional frequency space. Thus, a certain approximation to the required minimum point is obtained. Assume that we are in a unimodal vicinity of the global minimum, a clarification in the full four-dimensional space of coefficients by the gradient method is carried out.

As a result, we obtain the coefficients for models of (19) so that the standard deviation of the model from some sample  $A$  is minimal. For each model, the value of the combined selection criterion is calculated, and only those models that have the lowest values of this criterion are passed to the next level. The process continues until the minimum of the selected ensemble of criteria is reached. The nonlinear self-organization algorithm used to solve the prediction problem demonstrated quite high accuracy.

#### 4. Modification of the Volterra Neural Network Using the Method of Self-Organization

**4.1. The Volterra Network and Simulation Results.** In this section, an algorithm for building a dynamic object model is developed, and it can adequately set the initial values of the weight coefficients of a neural network, which significantly accelerates the learning process of neural network. Meanwhile, the algorithm of optimization of Volterra network structure is also considered.

The control of various dynamic objects usually involves the use of their mathematical models. In this case, when the model of a dynamic object is a priori unknown, it is necessary to build by using a neural network. Neural networks allow building models of investigated objects with a sufficiently high accuracy, but they require a long time to implement the learning process. When synthesizing control systems for dynamic objects especially various aircraft, the time for model building is limited. Therefore, the task of accelerating the work of a neural network is extremely important.

The main task of building and training a neural network in the case under study is approximation of a function. Based on a training sample of input data and function values, it requires to determine the weights of neural network, so that the result of the network (value of the output function) on the vector of input variables is as close as possible to the specified function value (training value) for this vector.

In the process of implementing the neural network training, the following procedures are performed in turn

for all input vectors: (1) The input vector values are passed through the network; the result of the network operation is found. (2) There is a deviation of the network result from the initial value. (3) The weights of connection of the links in network elements from the last layers to the first are changed. The change occurs in accordance with the gradient descent method. The goal is to find the minimum error for each element.

After that, the training condition for the end of algorithm is checked, i.e., how the performance of the neural network differs from the initial values. If the condition has not yet been fulfilled, then the algorithm returns to the second step. If the deviation from the original sample satisfies the conditions specified in the algorithm a priori, then the neural network is considered trained.

The method of self-organization is very similar to the neural network, but it is not the same. The method of self-organization determines the weights of connection using Gaussian normalization, and for each combination of functions a model of the form is constructed as

$$F_m^i = b_0 + b_1 F_k^{i-1} + b_2 F_l^{i-1} \quad (23)$$

where “ $i$ ” is the step number of the algorithm and “ $k$ ”, “ $l$ ”, “ $m$ ” are the function indices inside the “ $i$ ” and “ $i-1$ ” step sets of the algorithm, and the “ $k$ ” index should not coincide with “ $l$ ”.

In the transition from one step to the next, several best models are selected (in accordance with the Gabor principle). The combination continues as long as the error decreases using the test sample. After the algorithm is completed, it is required to go through all the steps of the algorithm in the reverse order and determine the weights of the basis functions.

$$\hat{F} = \hat{b}_0 + \hat{b}_1 F_1 + \dots + \hat{b}_N F_N \quad (24)$$

where  $\hat{F}$  represents result function;  $\hat{b}_i$  is final coefficient for the basis function; and  $F_i$  is the basis function.

Thus, in fact, the method of self-organization with the same structure as neural network is studied completely in different way. The first one is based on Gauss normalization method and the selection of best results, while the neural network is based on the method of back propagation and gradient descent method. The main disadvantage of neural network is the random selection of initial values of the weights, which leads to a long network training. From this aspect, the main task was to combine the advantages of the

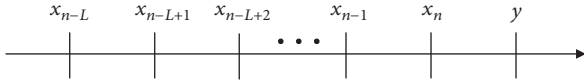


FIGURE 2: Input and output signals of Volterra network.

method of self-organization in the speed of work and the neural network in building a model of better approximation.

It is proposed to first search for an approximate minimum of error using the self-organization method, then to initialize the neural network weights of connection with the obtained values from self-organization method, and next, to find a more accurate approximation by neural network training. At the first stage, it is necessary to find a suitable network structure, which could easily be compared with the method of self-organization of all types of networks:

$$\Phi^i \neq \sum_k \Phi_k^{i-1} \omega_k \quad (25)$$

where  $\Phi_j^i$  denotes the function activation and  $\omega_k$  is connection weigh.

In the case of applying a function to the sum of the products of values of elements of the previous step on the weights of connections, it becomes difficult to initialize the weights of connections with values from the self-organization method. Similarly, it is difficult to distribute the weights if a chain of elements has several links with different weights.

As a result, the method of self-organization provides one weight for each basis function; it is not possible to only divide these weights into components. A type of neural network that has a suitable structure for combination with the method of self-organization is the Volterra network. This neural network allows using the result of the method of self-organization as a start point for learning a neural network. Accordingly, the weights coefficients of a function can be defined in the form as follows:

$$y = \sum_{i=0}^L x_{n-i} \left( w_i + \sum_{j=0}^L x_{n-j} \left( w_{ij} + \sum_{k=0}^L x_{n-k} (w_{ijk} + \dots) \right) \right), \quad (26)$$

$$\text{or } y = \sum_{i=0}^L x_{n-i} w_i + \sum_{i=0}^L \sum_{j=0}^L x_{n-i} x_{n-j} w_{ij} + \sum_{i=0}^L \sum_{j=0}^L \sum_{k=0}^L x_{n-i} x_{n-j} x_{n-k} w_{ijk} + \dots$$

Figure 2 shows the input and output signals of Volterra network. Here,  $x$  with indexes are the measuring signals, and they are indicated as the input vectors for the neural network;  $y$  is the output signal of the neural network;  $L+1$  is the dimension of the input vector.

If we expand the brackets in (26) and consider the Volterra network, then it is noticed that various combinations of products

$$x_{n-i} x_{n-j} x_{n-k} \dots \quad (27)$$

will duplicate each other; it is due to the fact that, in the case under study, when the factors are rearranged, their product does not change. For example,

$$\dots + w_{113} x_1 x_1 x_3 + \dots + w_{131} x_1 x_3 x_1 + \dots + w_{311} x_3 x_1 x_1 + \dots \quad (28)$$

where three terms of the expansion correspond to the same basis function. In the theory of Volterra it is stipulated that in such cases the corresponding weight coefficients “ $w$ ” should be equal, which means

$$w_{ijk} = w_{ikj} = w_{jik} = w_{jki} = w_{kij} = w_{kji} \quad (29)$$

In order to speed up the operation of the algorithm and to comply with the method of self-organization, it is necessary to combine these duplicate members. As a result of the operation of a complete neural network,  $N$  identical coefficients will be obtained with the same basis function. If we leave only one term of the expansion in consideration, then, under the same conditions, one coefficient is obtained, which has an expression as follows:

$$w_{ijk}^{(*)} = N w_{ijk} \quad (30)$$

The exclusion from the structure of the neural network of repeated basis functions allows greatly reducing the size of network and consequently the whole amount of calculations.

Examples are given for the value  $L=2$  (dimension of the input vector is 3, as shown in Figure 3). In this case, the number of input network elements will be  $3 + 9 + 27 = 39$ . And when  $L = 5$  (dimension of the input vector is 6), the number of input elements of the full network reaches  $6 + 36 + 216 + 1296 + 7776 + 46656 = 55986$ . It can be noticed that, even for such a small order, the number of input elements has been significant values.

To avoid repetition of the basis functions, the following method is constructed: the product is ordered by the indices of participating signals  $x$ ; in such a case, the basis function is not used more often than once. It is sufficient to generate functions in accordance with a simple algorithm. Firstly, all combinations for 0 are generated, then, the left ones without participation 0, after that for 1, the left ones without 0, 1 and for 2, etc. In Table 1 the rules for constructing nonrepeating index combinations for the Volterra network are shown.

The table discloses combinations for  $L = 4$  to the third level. Here each basis function is applied only once. For such a set of basic functions, it is possible to conduct initial training by the method of self-organization.

Thus, in order to use the self-organization method to accelerate the operation of the Volterra network, it is necessary to select the basis functions in a special way. The basis

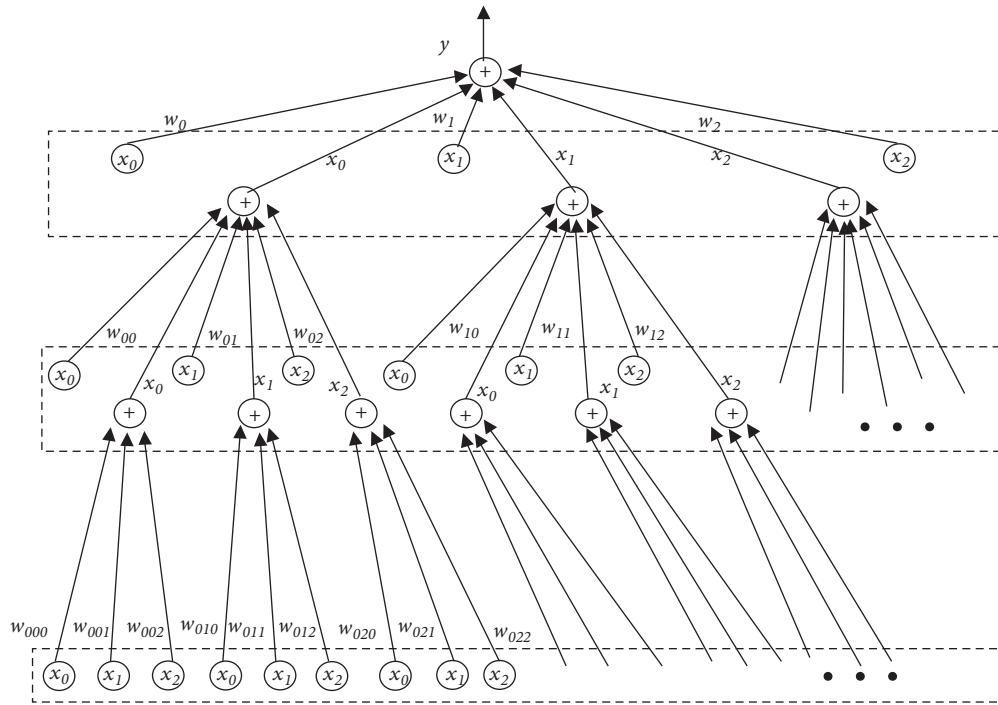


FIGURE 3: Volterra's neural network (for  $L = 2$ ).

TABLE 1: Rules for constructing non-repeating combinations of indices for a Volterra network.

1	2	3
0	00	000
1	01;11	001
2	02;12;22	002
3	03;13;23;33	003
		011;111
		012;112
		013;113
		022;122;222
		023;123;223
		033;133;233;333

functions used in the method of self-organization should be defined as follows:

$$\begin{aligned}
 &1, \\
 &x_0, x_1, x_2, \dots, x_L \\
 &x_0x_0, x_0x_1, x_0x_2, \dots, x_0x_L, x_1x_1, \dots, x_1x_L, x_2x_2, \dots, x_2x_L, \quad (31) \\
 &\dots, x_Lx_L, \\
 &\dots
 \end{aligned}$$

Such a set showed in (31) corresponds exactly to the set of products of signals  $x$  from (26). Then, if each product is used as a basis function in (24), then after the self-organization method is completed, the weight coefficients  $b_i$  will be obtained, which need to be assigned to the weight factors  $w$  in the first step of the Volterra network training. The correspondence of the pairs of coefficients

is determined, when creating the basic functions of self-organization method. And once the self-organization method is completed, the final coefficients are obtained already, it is possible to determine the corresponding basis functions and through them the weight coefficients of the Volterra network.

#### 4.2. The Modified Volterra Network and Simulation Results.

The structure of Volterra's network without the use of repetitive products is presented in Figure 4.

It should be noted that an additional element is introduced into the network structure, a constant, since this does not contradict the developed theory, at the same time, it gives full correspondence with the method of self-organization.

This reduction of the network structure leads to a sharp decrease in the number of input elements of the network. For the case  $L = 5$ , the number of input elements will be equal to  $6 + 21 + 56 + 126 + 252 + 462 = 923$ , this is significantly less than



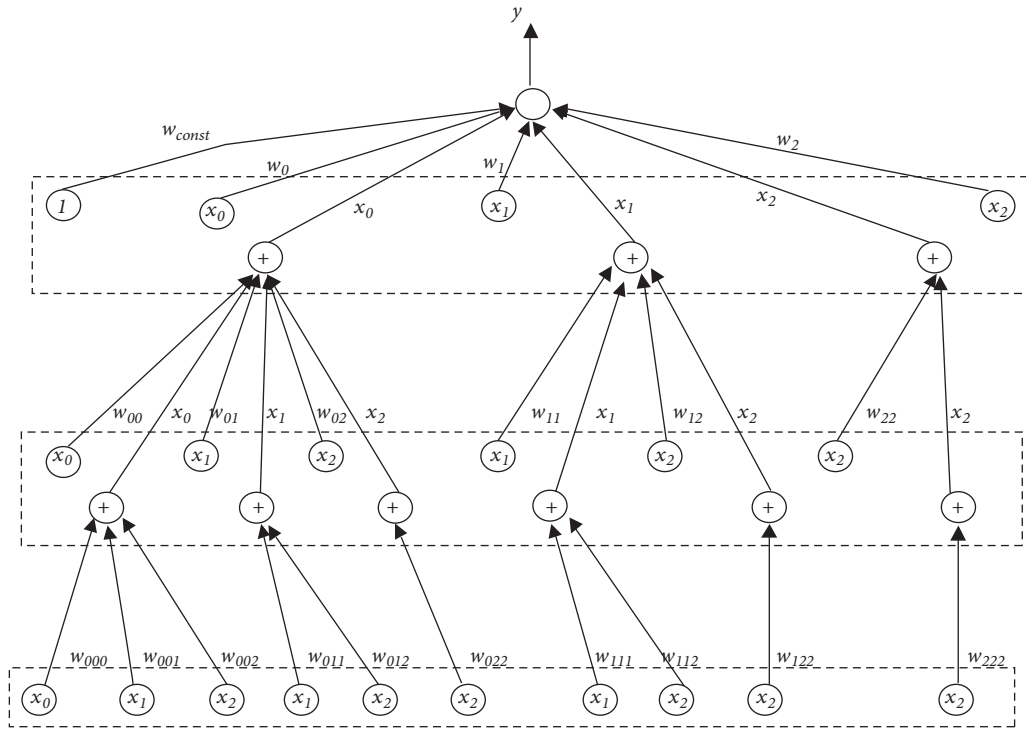


FIGURE 4: The reduced Volterra network.

the full set of functions. In addition, the reduced network does not require the complexity of the algorithm to control the equality of weights of connections of the coincident basis functions.

Based on the training sample with the help of a reduced neural network, it was possible to build a mathematical model (result of the network). It can be seen that, the accuracy of the model built by a reduced neural network is almost identical to the accuracy of the model made by an ordinary neural network. However, the speed of reduced neural network is significantly higher.

Thus, an algorithm for building a mathematical model based on a neural network is developed. To accelerate its work, it is proposed to determine the coefficients of the network by the method of self-organization. The Volterra neural network is represented and the reduced structure of this network is developed. Reduced neural network can significantly reduce the time to build a mathematical model.

Another approach that speeds up the process of building a model is parallelization of calculations in the implementation of a neural network. The operation of each layer of the neural network can be realized as a set of parallel threads in an amount equals to the product of the number of neurons of the current layer on the number of neurons of the previous layer.

Similarly, the neural network can also be parallelized for the error back propagation algorithm. The genetic algorithm involves step-by-step development of generations, evaluation of individuals of the current generation and the formation of a new generation of the best individuals of the previous one.

In parallel, it is impossible to calculate the next generation until the previous one is formed, but it is possible to work in parallel with individuals of the same generation. Parallel assessment of the quality of individuals of the current generation and the formation of the next generation of individuals help reduce the time of each cycle and the algorithm as a whole.

Considering that each network neuron can be calculated independently of the other neurons of its layer, as well as the fact that each individual neural network actively interacts with its parameters (synaptic weights) [20], the following parallelization method has been developed and implemented: in addition to simultaneous training of neural networks of one generation, each network is parallelized by neurons within a separate block.

The speed tests of the parallelized and sequential implementation of the system are performed; the results are shown in Table 2. In the process of testing, the same sets of neural networks of same structure are trained, using a sequential and parallel version of the training algorithm. Each neural network in the set contained 10 input, 15 hidden, and 2 output neurons.

It can be seen from Table 2 that the sequential implementation of the algorithm increases the system operating time in proportion to the number of trained neural networks. The increase time of system depending on the number of networks for sequential calculations is more than ten times greater than the same value for parallel calculations, which verifies the effectiveness of the developed method of parallelizing calculations and the feasibility of its application.

TABLE 2: Comparative characteristics of the speed of sequential and parallel implementation.

Type of calculation	Number of networks	Number of iteration	Number of training cycles	Work time
Sequential	1	10	1 000	0.51 s
	2	10	1 000	1.07 s
	3	10	1 000	1.66 s
Parallel	1	10	1 000	0.47 s
	2	10	1 000	0.53 s
	3	10	1 000	0.58 s

## 5. Verification of Algorithm for INS Errors Correction

In this section, an example of application and simulations are performed for well-known models of INS errors, and the INS is installed on a returning spacecraft to the atmosphere. As a test model, a typical error model of the platform INS is used in the following form:

$$\begin{aligned}
\delta \dot{v}_E &= -f_E - v_{up} \left( \frac{\delta v_E}{R} - u \sin \varphi \delta \varphi \right) + v_N \left( \frac{\delta v_E}{R} \operatorname{tg} \varphi \right. \\
&\quad \left. + u \cos \varphi \delta \varphi + \frac{v_E}{R} \sec^2 \varphi \delta \varphi \right) - \delta v_{up} \left( 2u \cos \varphi \right. \\
&\quad \left. + \frac{v_E}{R} \right) + \delta v_N \left( 2u \sin \varphi + \frac{v_E}{R} \operatorname{tg} \varphi \right) - f_{up} \cos \Phi_E \\
&\quad \cdot \sin \Phi_N + f_E \cos \Phi_N \cos \Phi_{up} \\
&\quad + f_N \left( \cos \Phi_N \sin \Phi_{up} + \sin \Phi_E \sin \Phi_N \cos \Phi_{up} \right) \\
&\quad - \sin \Phi_E \sin \Phi_N \sin \Phi_{up} + f_E \mu_E + B_E; \\
\delta \dot{v}_N &= -f_N - v_{up} \frac{\delta v_N}{R} - v_E \left( \frac{\delta v_E}{R} \operatorname{tg} \varphi + u \cos \varphi \delta \varphi \right. \\
&\quad \left. + \frac{v_E}{R} \sec^2 \varphi \delta \varphi \right) - \delta v_{up} \frac{v_N}{R} - \delta v_E \left( 2u \sin \varphi \right. \\
&\quad \left. + \frac{v_E}{R} \operatorname{tg} \varphi \right) + f_N \cos \Phi_E \cos \Phi_{up} - f_E \cos \Phi_E \sin \Phi_{up} \\
&\quad + f_{up} \sin \Phi_E + f_N \mu_N + B_N \\
\dot{\Phi}_E &= \left( -\frac{v_N}{R} - \frac{\delta v_N}{R} + \omega_E^{dr} \right) \cos \Phi_N + \frac{v_N}{R} \cos \Phi_{up} \\
&\quad - \left( \frac{v_E}{R} + u \cos \varphi \right) \sin \Phi_{up} + \left( \frac{v_E}{R} \operatorname{tg} \varphi + u \sin \varphi \right. \\
&\quad \left. + \frac{\delta v_E}{R} \operatorname{tg} \varphi + u \cos \varphi \delta \varphi + \frac{v_E}{R} \sec^2 \varphi \delta \varphi + \omega_{up}^{dr} \right) \\
&\quad \cdot \sin \Phi_N; \\
\dot{\Phi}_N &= \frac{v_E}{R} + u \cos \varphi + \frac{\delta v_E}{R} - u \sin \varphi \delta \varphi + \omega_N^{dr} \\
&\quad - \frac{v_N \sin \Phi_{up}}{R \cos \Phi_E} - \left( \frac{v_E}{R} + u \cos \varphi \right) \frac{\cos \Phi_{up}}{\cos \Phi_E} + \left( -\frac{v_N}{R} \right.
\end{aligned}$$

$$\begin{aligned}
&\quad \left. - \frac{\delta v_N}{R} + \omega_E^{dr} \right) \operatorname{tg} \Phi_E \sin \Phi_N - \left( \frac{v_E}{R} \operatorname{tg} \varphi + u \sin \varphi \right. \\
&\quad \left. + \frac{\delta v_E}{R} \operatorname{tg} \varphi + u \cos \varphi \delta \varphi + \frac{v_E}{R} \sec^2 \varphi \delta \varphi + \omega_{up}^{dr} \right) \operatorname{tg} \Phi_E \\
&\quad \cdot \cos \Phi_N
\end{aligned} \tag{32}$$

where  $v_N, v_E, v_{up}$  are velocity projections of the spacecraft on axis of the geographic trihedron,  $\delta v_N, \delta v_E, \delta v_{up}$  are, respectively, projections of errors in determining the velocity of spacecraft on axis of a geographic trihedron,  $\Phi_N, \Phi_E, \Phi_{up}$  denote the eviation angles between platform and geographic trihedron,  $f_N, f_E, f_{up}$  are projections of apparent acceleration of aircraft on axis of geographic trihedron,  $\omega_N^{dr}, \omega_E^{dr}, \omega_{up}^{dr}$  are projections of the drift velocity of gyro-stabilized platform on axis of geographic trihedron,  $\mu_N, \mu_E$  represent coefficient errors of accelerometer,  $B_N, B_E$  represent zero offsets of accelerometer,  $\varphi$  denotes local latitude,  $\delta \varphi$  denotes latitude error,  $u$  is Earth rotation speed, and  $R$  is Earth radius.

Then the error model of the northern channel of INS can be written as

$$\mathbf{x}_k = \mathbf{F} \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{33}$$

where

$$\begin{aligned}
\mathbf{x}_k &= \begin{bmatrix} \delta v_N \\ \Phi_E \\ \omega_E^{dr} \end{bmatrix}_k; \\
\mathbf{F} &= \begin{bmatrix} 1 & Tg & 0 \\ -\frac{T}{R} \cos \Phi_N & 1 & T \cos \Phi_N \\ 0 & 0 & 1 - T\beta \end{bmatrix}; \\
\mathbf{w}_k &= \begin{bmatrix} TB_N \\ 0 \\ TA \sqrt{2\beta} w \end{bmatrix}_k
\end{aligned} \tag{34}$$

Here  $w$  represents white noise, in the process of simulation and it is assumed that only the error in determining the velocity is obtained by measurement.

In Figure 5 the following notation is introduced: 1: measurements of a real INS; 2: model built by the neural network; 3: linear prior model; 4: nonlinear a priori model (33).

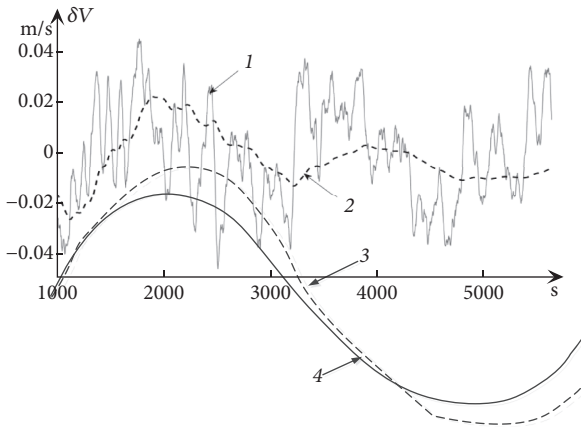


FIGURE 5: INS errors in determining velocity obtained with a real INS and by various models.

It can be seen that Figure 5 demonstrates the need to build a model of INS errors and the inexpediency of using a priori models for the correction of INS due to their low accuracy. In the process of modeling the reduced Volterra neural network model was built on a limited time interval. In practical applications, when correcting the INS from GPS, the interval is usually 1 second. Figure 5 presents the results of building a model for this time interval. The accuracy of the model built by the reduced neural network averages 85% of the nominal.

In Figure 6, 1 indicates the model of INS errors in determining the velocity; 2 indicates model of the reduced Volterra neural network, and in Figure 7, 1 indicates error of INS in determining the velocity; 2 indicates model built by Volterra's neural network without time constraints. The accuracy of building the model is on average 95% of the nominal.

From the results of simulations, we notice that the reduced Volterra network provides an acceleration of building models of a given accuracy, in comparison with the Volterra neural network on average by 7-10%. The accuracy of building a model in the correction interval averages 85% of the nominal.

### 6. Conclusions

This paper presents an advanced algorithmic method for increasing the accuracy of an INS of spacecraft. Three approaches for speeding up the work of neural network are suggested, which are extremely important in building mathematical models of the INS correction system. And the offline correction of INS is performed using the predictive error model constructed by the Volterra neural network modified by the self-organization algorithm. The modification of algorithm is validated to speed up the work efficiency of the neural network.

The accuracy of building a model using a reduced neural network is practically the same as the model built by an ordinary neural network. However, the speed of the reduced

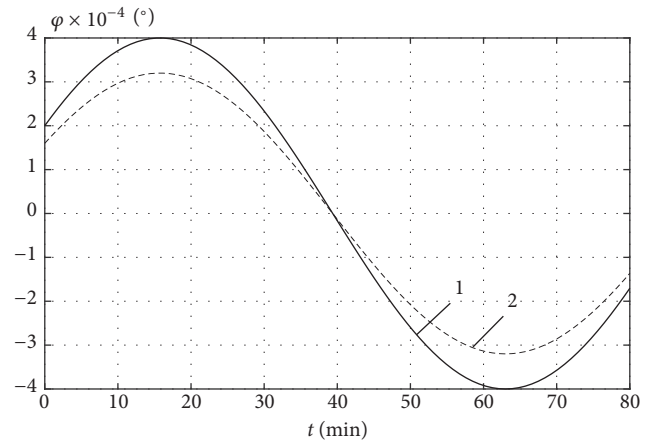


FIGURE 6: Angle of deviation of GPS from the plane of horizon and the model built by reduced Volterra neural network.

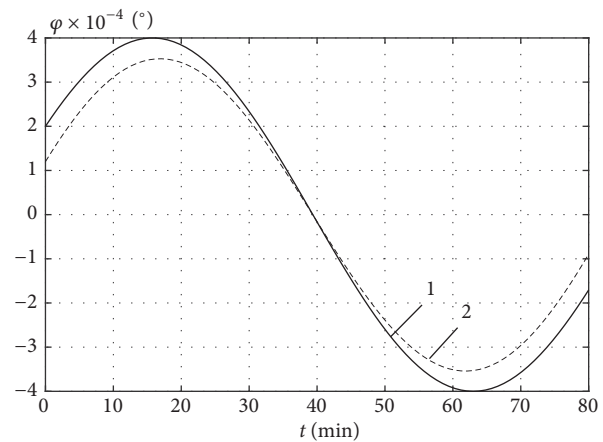


FIGURE 7: Angle of deviation of GPS from the plane of horizon and the model built by Volterra neural network.

neural network is significantly higher compared to a conventional neural network. In this paper an algorithm for building a mathematical model based on a neural network has been developed. To speed up its work, it is proposed to determine the network coefficients by self-organization, a Volterra's neural network is also presented, and a reduced structure of this network is developed. The reduced neural network can significantly reduce the time of building a mathematical model. Therefore, methods have been proposed to accelerate the operation of a neural network, which affect the process of mathematical models building of various dynamic objects, in particular, the INS error model of spacecraft. The simulation results show that the idea of combining neural network with navigation algorithm is feasible and has a wide application prospect, and the prospects for further research are related to the development of algorithms for constructing models with desired properties, for example, models with enhanced characteristics of observability, identifiability, and sensitivity, etc.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

There are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work is supported by the “Intelligent Ammunition System Theory and Key Technology Innovation Induction Base” founded by Chinese Ministry of Education.

## References

- [1] M. Y. Ovchinnikov, D. I. Penkov, D. S. Roldugin, and D. S. Ivanov, “Magnetic orientation systems for small satellites,” *Keldysh Institute of Applied Mathematics*, 2016, article no 366.
- [2] M. S. Selezneva and K. A. Neusypin, “Development of a measurement complex with intelligent component,” *Measurement Techniques*, vol. 59, no. 9, pp. 916–922, 2016.
- [3] E. L. Akim, A. P. Astakhov, R. V. Bakit’ko et al., “Autonomous navigation system of near-Earth spacecraft,” *Journal of Computer and Systems Sciences International*, vol. 48, no. 2, pp. 295–312, 2009.
- [4] C. Guo, F. Li, Z. Tian, W. Guo, and S. Tan, “Intelligent active fault-tolerant system for multi-source integrated navigation system based on deep neural network,” *Neural Computing and Applications*, pp. 1–18, 2019.
- [5] H. Myung and H. Bang, “Spacecraft parameter estimation by using predictive filter algorithm,” *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 3452–3457, 2008.
- [6] H. M. Romero Ugalde, J. Carmona, V. M. Alvarado, and J. Reyes-Reyes, “Neural network design and model reduction approach for black box nonlinear system identification with reduced number of parameters,” *Neurocomputing*, vol. 101, pp. 170–180, 2013.
- [7] C. Jiang, Y. Chen, S. Chen et al., “A mixed deep recurrent neural network for MEMS gyroscope noise suppressing,” *Electronics*, vol. 8, no. 2, pp. 181–195, 2019.
- [8] H. Liu and J. Wang, “Integrating independent component analysis and principal component analysis with neural network to predict Chinese stock market,” *Mathematical Problems in Engineering*, vol. 2011, Article ID 382659, 15 pages, 2011.
- [9] C. W. Chen and Y. C. Chang, “Support vector regression and genetic algorithm for HVAC optimal operation,” *Mathematical Problems in Engineering*, vol. 2016, Article ID 6212951, 10 pages, 2016.
- [10] Q. Zhu, Y. Han, C. Cai, and Y. Xiao, “Robust optimal navigation using nonlinear model predictive control method combined with recurrent fuzzy neural network,” *Mathematical Problems in Engineering*, vol. 2018, Article ID 8014019, 19 pages, 2018.
- [11] K. Asadi, P. Chen, K. Han, T. Wu, and Lobaton, “Real-time Scene segmentation using a light deep neural network architecture for autonomous robot navigation on construction sites,” in *Proceedings of the The ASCE International Conference on Computing in Civil Engineering*, 2019.
- [12] W. Zhou, X. Zhu, J. Wang, and Y. Ran, “A new error prediction method for machining process based on a combined model,” *Mathematical Problems in Engineering*, vol. 2018, Article ID 3703861, 8 pages, 2018.
- [13] G. P. Liu, V. Kadiramanathan, and S. A. Billings, *On-Line Identification of Nonlinear Systems Using Volterra Polynomial Basis Function Neural Networks*, Elsevier Science Ltd, 1998.
- [14] Y.-S. Yang, W.-D. Chang, and T.-L. Liao, “Volterra system-based neural network modelling by particle swarm optimization approach,” *Neurocomputing*, vol. 82, pp. 179–185, 2012.
- [15] O. E. Bukharov and D. P. Bogolyubov, “Parallelization of self-learning decision support system based on neural networks and a genetic algorithm,” *System Administrator*, vol. 9, no. 142, pp. 88–92, 2014.
- [16] O. E. Bukharov and D. P. Bogolyubov, “Development of a hybrid decision support system and its application,” *Devices and Systems. Management, Control, Diagnostics*, vol. 1, pp. 25–33, 2018.
- [17] O. E. Bukharov and D. P. Bogolyubov, “Development of a decision support system based on neural networks and a genetic algorithm,” *Expert Systems with Applications*, vol. 42, no. 15-16, pp. 6177–6183, 2015.
- [18] W. Qin and W. Yong, “The anti-wind up adaptive control of aircraft based on neural networks,” *Aerospace Control*, vol. 29, no. 3, pp. 60–64, 2011.
- [19] L. Jian, L. Gui-Fang, and H. Shu-Yu, “Neural networks based model reference adaptive dynamic inversion flight control,” *Science Technology and Engineering*, vol. 12, no. 19, pp. 4716–4720, 2012.
- [20] Z. Min and X. Qi-hua, “The improved BP neural network model reference adaptive control,” *Computer Engineering & Software*, vol. 17, pp. 118–123, 2015.
- [21] B. Subudhi and D. Jena, “A differential evolution based neural network approach to nonlinear system identification,” *Applied Soft Computing*, vol. 11, no. 1, pp. 861–871, 2011.
- [22] W. F. Xie, Y. Q. Zhu, Z. Y. Zhao, and Y. K. Wong, “Nonlinear system identification using optimized dynamic neural network,” *Neurocomputing*, vol. 72, no. 13-15, pp. 3277–3287, 2009.
- [23] K. A. Neusypin, “Modern systems and methods of guidance, navigation and control of aircraft,” *MGOU*, 2009, article no 500.
- [24] A. G. Ivakhnenko and J. Ya. Müller, “Self-organization of predictive models,” *Kiev. Equipment*, 1985.

Copyright © 2019 Danhe Chen et al. This is an open access article distributed under the Creative Commons Attribution License (the “License”), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License. <https://creativecommons.org/licenses/by/4.0/>