IJCTA, 9(30), 2016, pp. 61-71

Study of Type I and Type II Errors Occurring in the Automatic Fault Detection after Remote Reconfiguration of FPGA of a Mobile Robot for Space Application

Valery Dmitrievich Ivchenko* Aleksandr Sergeyevich Ignatov* Petr Germanovich Krug* Maksim Vyacheslavovich Kurakov* and Sergey Aleksandrovich Pavelyev*

Abstract : We consider the problem of ensuring the reliability of the procedure of remote modification of intelligence of mobile robots for space application, implemented on the basis of FPGA. We propose a functional structure of the reconfiguration controller, which performs the neural network-based classification of faults occurring during remote modification of intelligence of mobile robots for space application. We consider the software and algorithmic support of the reconfiguration controller, which provides detection of certain kinds of occurring faults, the place where they occur and the fact of occurrence of several faults, as well as the implementation of periodic diagnostics in the process of operation of mobile robots for space application. We investigate the likelihood of type I and type II errors in the course of automatic fault detection.

Keywords : Space-based mobile robot, remote modification, field programmable gate array (FPGA), hardware reconfiguration, neural network-based classification.

1. INTRODUCTION

Large duration of the deep space exploration missions, which is measured in months and years, results in a number of specific requirements to the design of mobile robots for space application [1-2]. For example, mobile robots for space application require a high level of survivability (degradation), the ability to perform tasks in the conditions of failure of some mechanical components, a part of hardware or software, as well as in the conditions of previously unforeseen changes in the environment parameters. It is also required to ensure a high level of versatility (reserve), the ability of mobile robots for space application to continue successful functioning under changing task objectives (mission change) or under changing ways of achieving them.

It is possible to satisfy these requirements, provided the artificial intelligence of mobile robots for space application will be able to be changed (modified) when new tasks are posed before them or the conditions for their implementation worsen. Of particular relevance in this context is the use of hardware and software with reconfigurable structure implemented on the basis of field programmable gate array (FPGA). The use of these technologies allows remotely transmitting (from a planet-based complex or from the Earth) new or modified software modules into the control device of mobile robots for space application with automatic firmware renewal of the mobile robot's FPGA.

Much attention in this case should be paid to ensuring the reliability of the reconfiguration procedure. A topical direction in this area is the introduction of automated or automatic means of search and localization of faults [3-4]. To solve this problem, specialized software and algorithmic support has been developed for the reconfiguration controller, which performs the neural network-based classification of faults occurring during remote modification of intelligence of mobile robots for space application.

* Moscow Technological University (MIREA) 78 Vernadsky Avenue, Moscow 119454, Russia

The article discusses the problem of application of the neural network-based classification technologies to provide reliable detection of the type of occurring malfunctions, their place and the fact of occurrence of several faults, as well as to implement periodic diagnostics in the process of operation of mobile robots for space application.

In this connection, an important task is to study the probability of type I and type II errors during neural network-based classification of faults [5-7]. A type I error, also known as a false alarm, false response or false positive actuation, is a situation when the neural network classifies the state of the circuit as belonging to one of the types of faults, while the input vector of the neural network corresponds to the circuit state in the absence of faults. A type II error, which is also called the omission of event or false negative actuation, is a situation when the neural network classifies the state of the circuit as operable, while the input vector of the neural network corresponds to a circuit state in the presence of a particular type of fault.

In the diagnostics of the presence of faults in the circuit implementing the intelligence of mobile robots for space application, the most important task is to minimize the number of missed faults (type II errors), because in this case the continued operation of the robot as a whole is threatened. However, in the condition of realization of deep space exploration missions, it is also important to prevent a significant increase in the risk of type I errors, because the implementation of the reconfiguration procedure to remove a falsely detected error is connected with additional time and energy costs.

2. METHOD

2.1. The functional structure of the hardware-reconfigurable digital module of intelligent control of mobile robots for space application

Reconfiguration of FPGA is carried out on the initiative of the remote support center on the Earth under changing the mission objectives of mobile robots for space application or changing the ways of achieving them [8-9]. Reconfiguration can also be initiated by the mobile robot itself to repair the faults discovered by it in the course of self-diagnostics [10-11].

A system of intelligent control of mobile robots for space application is proposed on the basis of hardware-reconfigurable digital platform (FPGA), which includes the following equipment [12]:

1. Reconfiguration controller performing neural network-based classification of faults occurring during remote modification of intelligence of mobile robots for space application;
2. Programmable hardware on the basis of FPGA for the strategic, tactical and drive levels of the mobile robot control;
3. Reconfiguration server which is a part of a remote support center connected with the mobile robot by a network infrastructure and which is designed for remote control of a group of mobile robots for space application, including for implementing connected reconfiguration of their software and hardware.

In accordance with the presented hardware composition, we propose a functional structure of hardware-reconfigurable digital module of intelligent control of mobile robots for space application (Figure 1).

The self-test tools perform the following functions:

1. Providing diagnostics of the functional state of the mobile robot intelligence after modifications in automatic mode;
2. Testing the operability of the reconfigurable module;
3. Determination of the type of fault;
4. Localization of the fault and the formation of quarantine around the faulty circuit elements;
5. Determination of the type of fault under the possible occurrence of more than one fault;
6. Notification of the remote support center concerning the results of diagnostics;
7. Repeated reconfiguration to repair the fault;
8. Ensuring the monitoring of the functional state of circuits that implement mobile robot intelligence in real time.

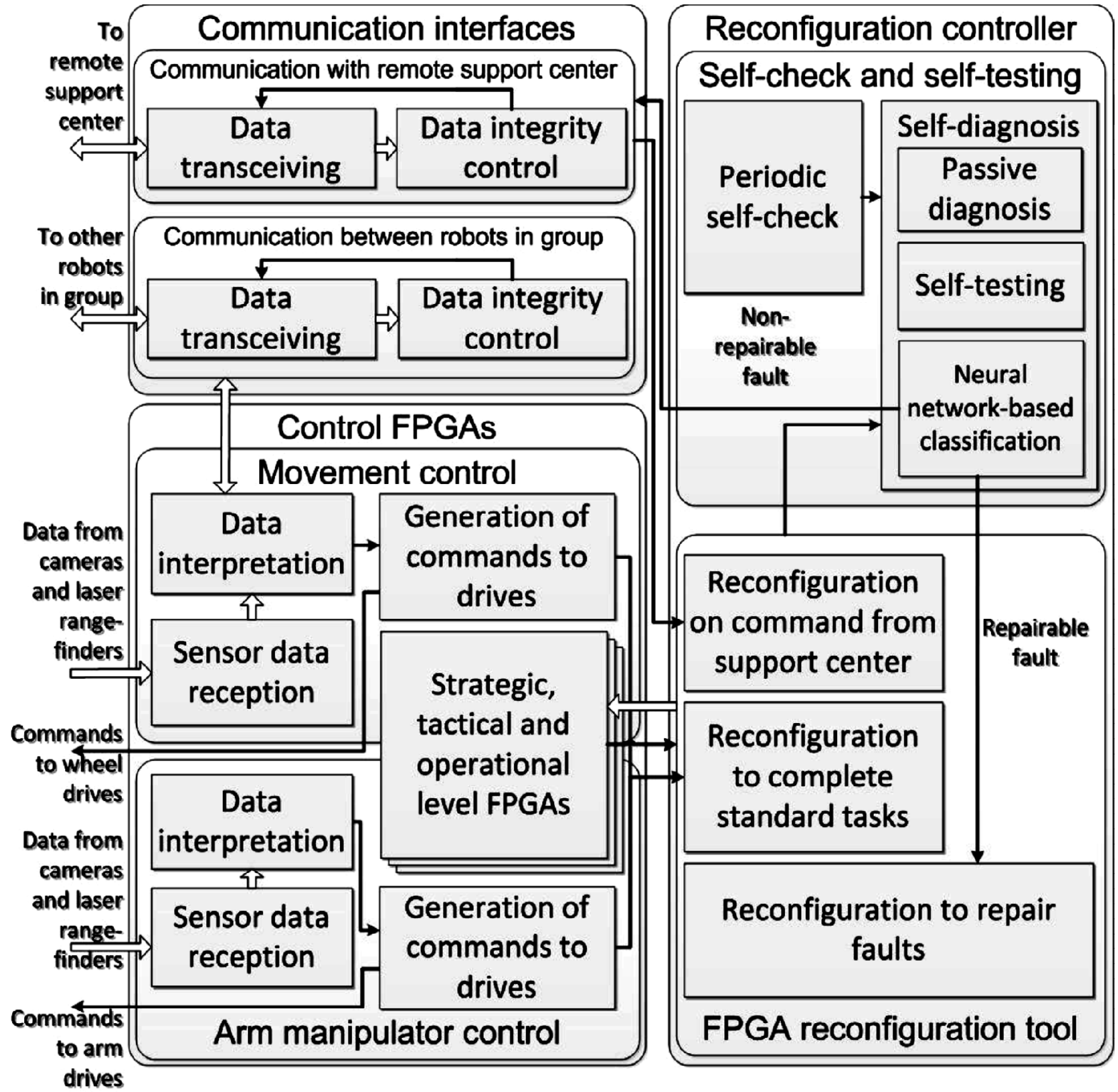

Fig. 1. Functional structure of hardware-reconfigurable digital module of intelligent control of mobile robots for space application.

2.2. Structure of the neural network-based fault classifier

The described self-test tasks can be performed by both a neural network-based classifier consisting of a single neural network and a neural network-based classifier consisting of four independent neural networks.

The selection of one of these two variants of realization of a neural network-based classifier can be carried out based on evaluation of the computational complexity of hardware implementation of the relevant structures of the neural network-based classifier.

To evaluate the computational complexity of the neural network, this network can be represented as a hierarchical structure of the following type [13]:

$$y^s = \psi^{(M,1)}(\varphi^{(M,1)}(w^{(M,i)}, (\psi^{(M,-1,i)}(\ldots))), \psi^{M(0,i)} = x_i^s, \tag{1}$$

$$w^{(\mu,i)} = \{w_j^{(\mu,i)}\}, i = 1, 2, \ldots, N_{\mu-1}, \mu = 1, 2, \ldots M,$$

where X = <x, y> is a collection of S precedents concerning the dependence $y(x)$, $x = \{x^s\}$, $y = \{y^s\}$, $s = 1, 2, \ldots S$ which are characterized by a set of N input characteristics $\{x_j\}$, $j = 1, 2, \ldots, N$, where j is the number of the characteristic, and an output characteristic y, $< x^s, y^s >$, $x^s = \{x_j^s\}$ is the s-th precedent, where x_j^s is the value of the j-th input and y^s is the value of the output characteristic for the s-th precedent (exemplar) of the sample, $y^s \in \{1, 2, \ldots, K\}$, where K is the number of classes, K > 1, M is the number of layers,

N_μ is the number of nodes (neurons) in the μ-th layer,

$\varphi^{(\mu, i)}$ is the discriminant (weight) function of the μ-th node in the μ-th layer,

$\psi^{(\mu, i)}$ is the activation function of the *i*-th node in the μ-th layer,

$w_j^{(\mu, i)}$ is the weight value of the *j*-th input of the *i*-th node in the μ-th layer.

The structural complexity of a neural network model, whose main structural elements are neurons, can be characterized by the number of neurons N_n, which can be determined for a multilayer neural network by the formula [14]:

$$N_n = \sum_{\mu=1}^{M} N_\mu \tag{2}$$

The computational complexity of the *i*-th neuron in the μ-th layer can be defined as :

$$T^{(\mu, j)} = N^{(\mu, i)}(T_c^{(\mu, i)} + T_\varphi^{(\mu, i)}) + T_\psi^{(\mu, i)}, \tag{3}$$

where $N(\mu, i)$ is the number of inputs of the *i*-th neuron in the μ-th layer;

$T_c^{(\mu, i)}$ is the computational complexity of one synapse of the *i*-th neuron in the μ-th layer;

$T_\varphi^{(\mu, i)}$ is the computational complexity of the discriminant function of the *i*-th neuron in the μ-th layer for processing two arguments;

$T_\psi^{(\mu, i)}$ is the computational complexity of the activation function of the *i*-th neuron in the μ-th layer.

The computational complexity of synapses, discriminant functions and activation functions are determined, taking into account the conditions of hardware implementation of the neural network.

The computational complexity of the feedforward neural network under parallel implementation of calculations (for example, under hardware implementation of the neural network on FPGA) will be defined by the formula:

$$T = \sum_{\mu=1}^{M} \max_{i=1,2,\dots N_\mu}\{T^{(\mu, i)}\} \tag{4}$$

The logical transparency of the neural network largely depends on the total number of connections in the neural network and the number of links that connect specific neurons (the number of inputs of the neurons in the inner layers). The fewer the links, the simpler the neural network is, and so it is more convenient for analysis and interpretation.

The sparsity coefficient of the links of the feedforward neural network is determined by the formula:

$$K_R = \frac{N_{w=0}}{\sum_{\mu=1}^{M} N_{(\mu-1)} N_\mu}, \; N_0 = N \tag{5}$$

where $N_{w=0}$ is the number of the neural network's weights equal to zero.

The connectivity coefficient of the multilayer neural network is defined as:

$$K_C = 1 - K_R = 1 - \frac{N_{w=0}}{\sum_{\mu=1}^{M} N_{(\mu-1)} N_\mu} \tag{6}$$

The average connectivity coefficient K_M of the feedforward neural network shows the average number of inputs of the neurons of all layers except the first one:

$$K_M = \frac{1}{\sum_{\mu=2}^{M} N_\mu} \sum_{\mu=2}^{M} \sum_{i=1}^{N_\mu} (N^{(\mu, i)} - N_{w=0}^{(m, i)}) \tag{7}$$

where $N_{w=0}^{(\mu,i)}$ is the number of weights of the i-th neuron in the μ-th layer that are equal to zero.

The more unit synapses are in the neural network (the connections whose weights equal one in absolute value), the easier its implementation is (especially, the hardware one) and the easier analysis and interpretation are [15-16].

The share of unit synapses in the feedforward neural network is calculated by the formula:

$$K_L = \frac{N_{w=1}}{\sum_{\mu=1}^{M} N_{(\mu-1)} N_\mu}, N_0 = N \tag{8}$$

where $N_{w=1}$ is the number of weights in the neural net that are equal to one in absolute value.

The share of non-unit synapses K_N in the feedforward neural network is determined by the expression:

$$K_N = 1 - K_L = 1 - \frac{N_{w=1}}{\sum_{\mu=1}^{M} N_{(\mu-1)} N_\mu}, N_0 = N \tag{9}$$

Since the logical transparency of the neural network connections depends largely on the sparseness and simplicity of connections, it is characterized by a coefficient showing the share of the binary (zero or one in absolute value) weights among the total number of weights of the neural network [17-19].

The coefficient of logical transparency of the feedforward neural network connections is determined by the expression:

$$K_T = K_R + K_L = \frac{N_{w=1} + N_{w=1}}{\sum_{\mu=1}^{M} N_{(\mu-1)} N_\mu}, N_0 = N \tag{10}$$

The coefficient K_S of logical non-transparency (fuzziness) of the feedforward neural network connections is calculated as $K_S = 1 - K_T = 1 - (K_R + K_L)$.

For more accurate evaluation of the logical transparency, one can define logical transparency of a neural network in terms of the logical transparency of its elements.

The coefficient of logical transparency of the i-th neuron in the μ-th layer of the neural network $K_E^{(\mu,i)}$ is determined by the type of the used activation function. For the linear and threshold activation functions: $K_E^{(\mu,i)} = 1$, for all other activation functions $K_E^{(\mu,i)} = 0$ [20-21].

The logical transparency coefficient of a multilayer neural network will be calculated as:

$$K_U = \frac{\sum_{\mu=1}^{M} \sum_{i=1}^{N_\mu} K_E^{(\mu,i)}}{K_M \sum_{\mu=1}^{M} N_\mu}, K_M \neq 0 \tag{11}$$

The larger the coefficient K_U, the higher is the logical transparency level of the neural network, and conversely, the smaller the , the lower is the level of logical transparency of the neural network.

One of the most important characteristics of neural network models is the quality of approximation. The approximation quality for the same error level is higher, when the number of the used weights is smaller [22-24].

The approximation quality coefficient of a neural network model is defined as the average proportion of errors attributable to non-zero weights of the neural network [25-27]:

$$K_A = \frac{E}{N_w - N_{w=0}} \tag{12}$$

where E is the total error made by the neural network (for example, the mean-square error). As the error E, one can use the neural network training error (computed by the learning sample) or the work error of the neural network (calculated by the test sample).

It follows from the presented expressions that the computational complexity of the neural network grows faster when the number of connections between the neurons of the inner layers increases; whereas for the hardware implementation of neural network on FPGA, the total computational complexity of the neural network will be determined by the maximum computational complexity of particular neurons. The logical transparency coefficient of a multilayer neural network is inversely proportional to the product of the number of neurons in the adjacent layers, whereas the approximation quality coefficient is inversely proportional to the number of neurons with nonzero weights. Consequently, the use of a neural network with a complex structure, designed for determination of a large number of classes of faults (*i.e.*, containing a large number of neurons in the output layer) will be significantly more costly in terms of computing resources and will have a lower coefficient of approximation quality in comparison with using the set of four independent neural networks solving the limited classification problems on the selected stages of the self-test procedure.

Thus, the most effective and advantageous solution is the implementation of a composite structure of neural network-based classifier of faults occurring during remote modification of the intelligence of mobile robots for space application, consisting of several neural networks (Figure 2):

- a neural network determining the type of fault (NN1);
- a neural network determining the place of occurrence of the fault (NN2);
- a neural network determining the fact of occurrence of more than one fault (NN3);
- a neural network of periodic diagnostics, operating in the online mode, after the transfer to the FPGA of the control over the mobile robot for space application (NN4).

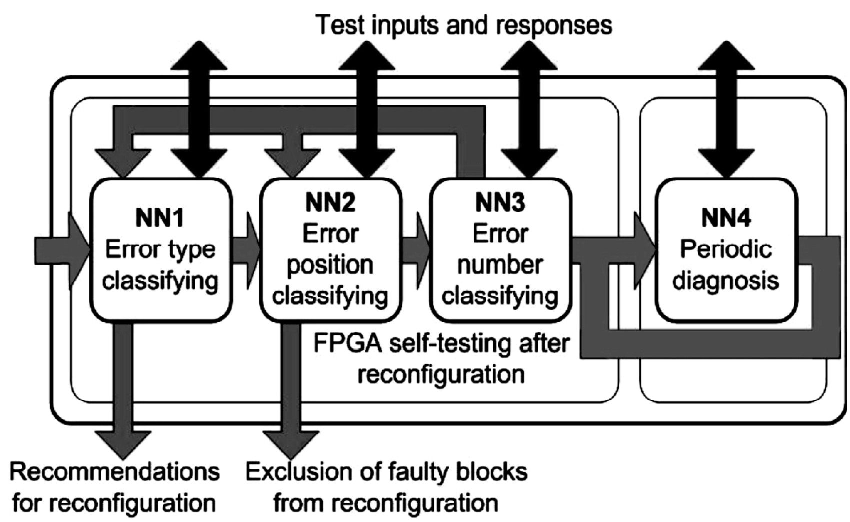

Fig. 2. Structure of the fault classifier consisting of four neural networks.

A generalized algorithm of fault classification is shown in Figure 3.

3. RESULTS

Consider the results of studying the operation of a neural network-based classifier in terms of its tolerance to type I and type II errors.

To evaluate the tolerance of each neural network to the emergence of a type I error, simulation is carried out of the neural network operation when applying to its input test vectors not participating in training and corresponding only to the situations when there are no faults in the diagnosed scheme.

To evaluate the tolerance of each neural network to the occurrence of a type II error, the simulation is carried out of the neural network operation when applying to its input the test vectors not participating in training and corresponding only to the situations when there is a fault of a certain type in the diagnosed scheme.

The graph of dependence of the probability of type I and type II errors on the number of neurons in the inner layer of NN1 is shown in Figure 4.

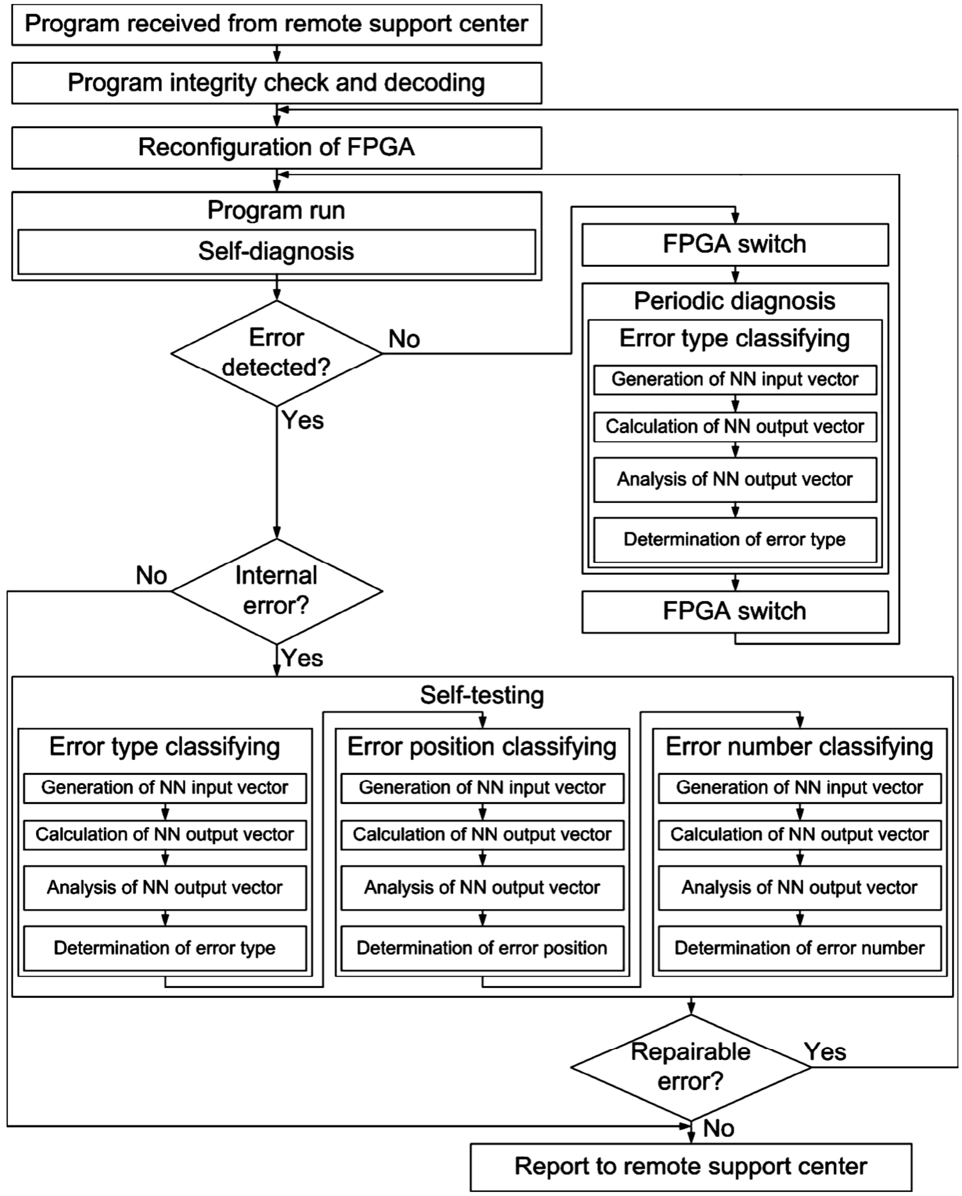

Fig. 3. Structure of the fault classifier consisting of four neural networks.

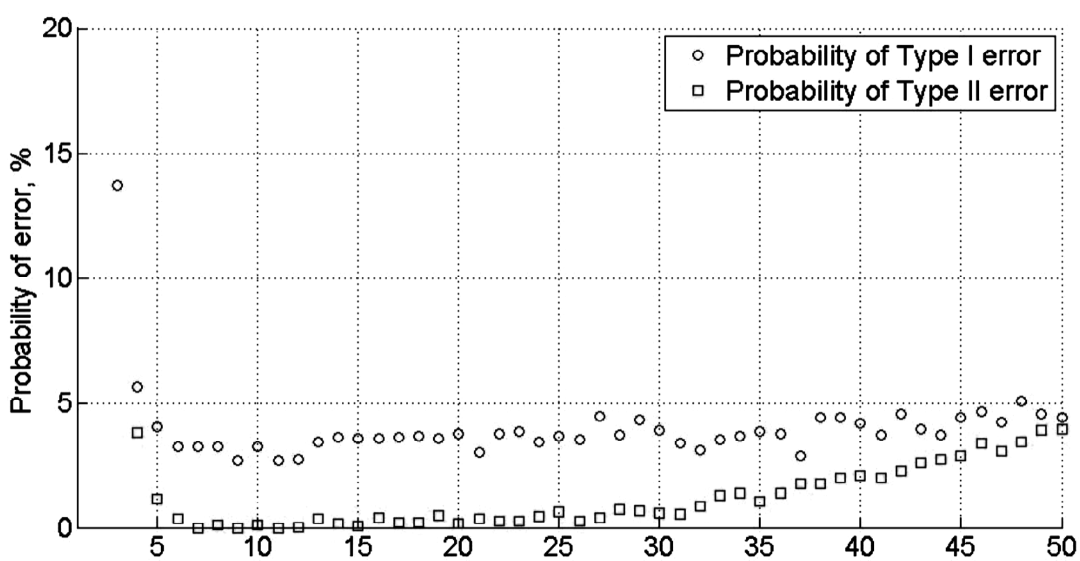

Fig. 4. Dependence of the probability of type I and type II errors on the number of neurons in the inner layer of NN1.

It can be seen from the graph that NN1 demonstrates higher resistance to the emergence of a type II error than to a type I error. Given the greater significance of the former for the problem being solved, it is, of course, a positive feature of the neural network determining the type of fault.

The graph of dependence of the probability of type I and type II errors on the number of neurons in the inner layer of NN2 is shown in Figure 5.

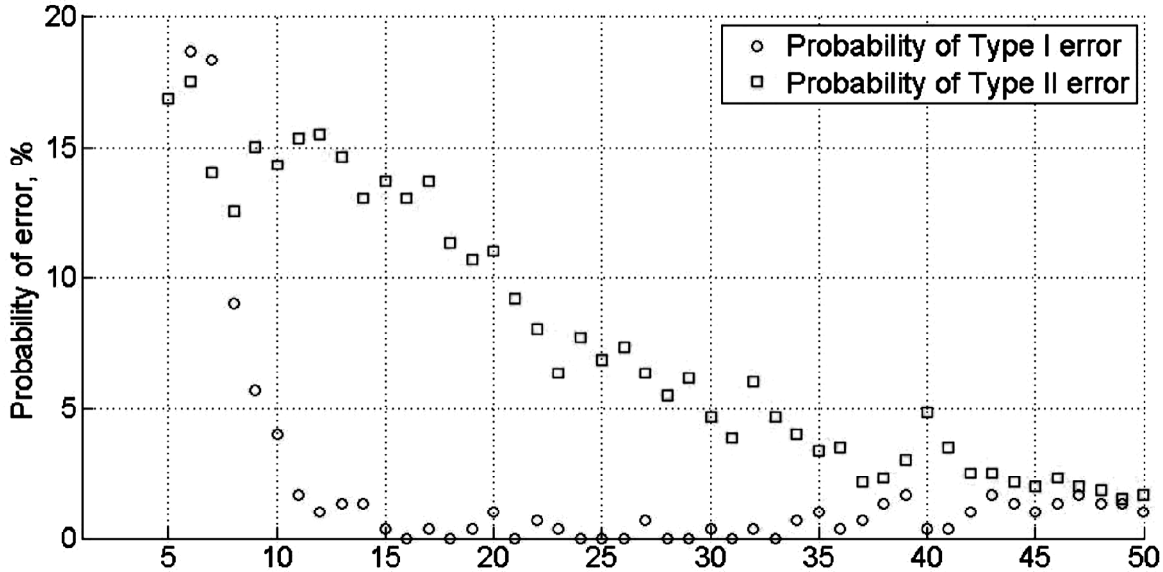

Fig. 5. Dependence of the probability of type I and type II errors on the number of neurons in the inner layer of NN2.

The graph shows that NN2 demonstrates higher resistance to a type I error than to a type II error. A higher probability of the occurrence of a type II error is associated with a large number of neurons in the output layer of NN2 as compared with NN1.

The graph of dependence of the probability of type I and type II errors on the number of neurons in the inner layer of NN3 is shown in Figure 6.

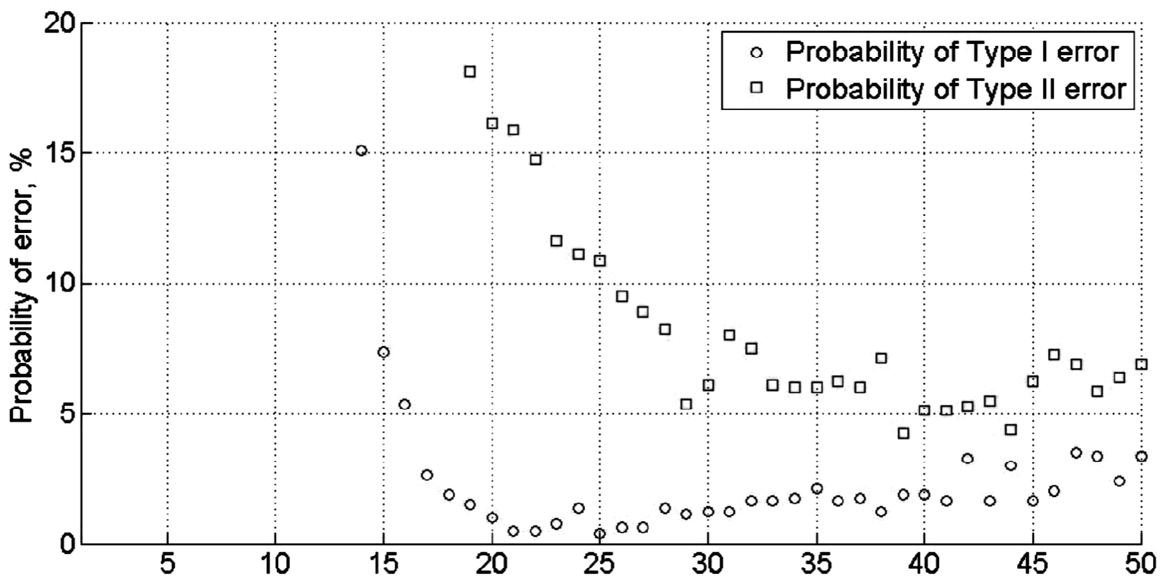

Fig. 6. Dependence of the probability of type I and type II errors on the number of neurons in the inner layer of NN3.

It can be observed in the graph that NN3 also exhibits higher resistance to a type I error than to a type II error. A higher probability of the occurrence of type I or type II errors is related to the specifics of classification of the fact of occurrence of more than one fault (to each class, there can correspond more than one active neuron in the output layer of the neural network).

Based on the simulation results, we can conclude that the total probability of a type I or type II error arising in the classification of faults, the place of their occurrence and the fact of occurrence of multiple faults is rather large, and its value cannot be neglected.

To improve the reliability of classification, we propose to evaluate the result of work of NN1, NN2 and NN3 according to the following principle: the classification is considered successful and its result is transferred for further analysis, provided all three neural networks confirm the presence or absence of a fault of a particular type in the tested scheme, otherwise, the result of classification is considered unreliable and, to determine the circuit state, it is required to apply additional methods of testing, such as deterministic methods of exhaustive search of input actions, etc.

With this approach, considering the probability of type I and type II errors in the operation of NN1, NN2 and NN3 independent of one another, the total probability of an error of type I or II will be calculated from the expression:

$$P_S = P_1 * P_2 * P_3, \tag{13}$$

where P_1 is the probability of occurrence of an error of type I or II in the operation of NN1,

P_2 is the probability of occurrence of an error of type I or II in the operation of NN2,

P_3 is the probability of occurrence of an error of type I or II in the operation of NN3.

The graph of dependence of the total probability of an error of type I or II, in accordance with the expression (13), is shown in Figure 7.

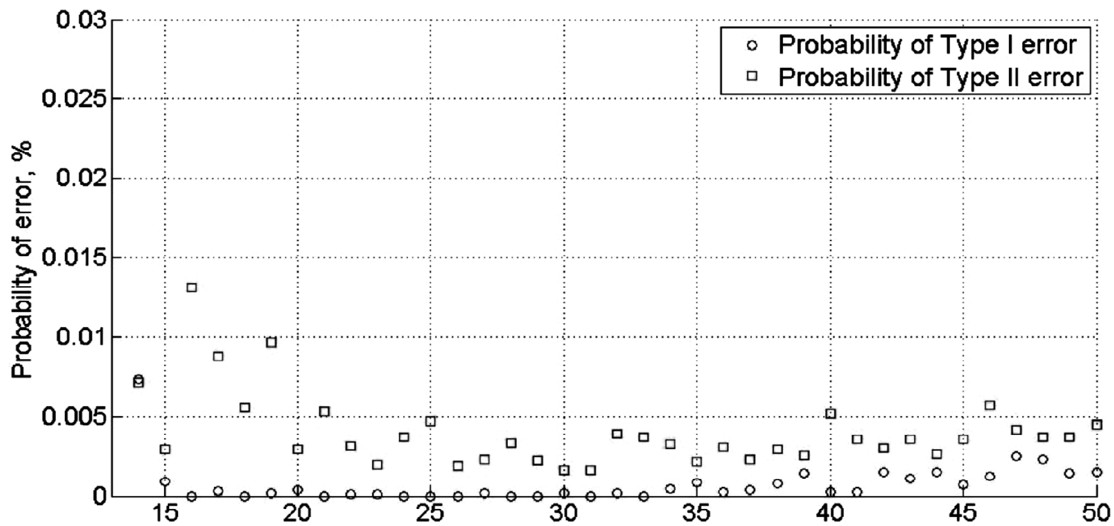

Fig. 7. Dependence of the total probability of type I or II errors on the number of neurons in the inner layer of neural networks included in the neural network-based classifier.

It can be concluded from the above graph that, when evaluating the performance of a neural network-based classifier according to the proposed principle, the value of the total probability of errors of type I or II is within an acceptable range for the class of problems being solved [5-7].

Neural network NN4 of the periodic diagnostics of the circuit state

For the periodic diagnostics of the state of the circuit, it is proposed to use a neural network that determines the type of the arising fault (similar to NN1). Thus, such diagnostics will not require significant time and energy costs and may be carried out at predetermined periods of operation of mobile robots for space application, not requiring carrying out critical control tasks, for example, during battery recharging.

The periodic diagnostics of the state of the circuit, after the transfer of control to it, is intended to provide additional control over the appearance of malfunctions during operation. The periodic diagnostics should solve the problem of revealing the fact of malfunction. The problem of localization of the place of a fault or faults must be solved within the framework of a self-test procedure which is launched after the transfer of control to the backup FPGA on the basis of detection of a fault at the stage of periodic diagnostics.

In this case, it is important that the periodic diagnostics should not require considerable time and energy costs, because generally it needs to be carried out substantially more often than the self-test procedure.

Based on these considerations, for the periodic diagnostics of the circuit condition, it is proposed to use a neural network that determines the type of the occurred fault that has the simplest structure and exhibits higher resistance to the emergence of type II errors than type I errors. When an error is detected at the stage of periodic diagnostics, a self-test procedure should be performed for the circuit that provides resistance to the errors of type I and II.

Periodic diagnostics can be carried out at certain stages of functioning of mobile robots for space application that do not require implementation of critical control tasks, for example, while recharging the batteries.

4. CONCLUSION

The proposed structure of the hardware-reconfigurable digital module of intelligent control of mobile robots for space application is designed to enhance survivability and autonomy of mobile robots by using the technologies of remote modification of intelligence based on reconfiguration of hardware, implemented on the basis of FPGA. The developed concept combines the use of methods and technologies of artificial intelligence and the reconfiguration capabilities of the FPGA hardware platform, including partial reconfiguration in real time as a means to deal with the environment uncertainty. The practical implementation of this concept involves selective use of certain technologies of knowledge processing depending on the specifics of the problems being solved, the features of the object being controlled, its functional purpose, operating conditions, etc. The application of the developed technologies will enhance the efficiency of use of mobile robots for space application and, as a consequence, reduce the costs of the space missions.

The developed structure of the reconfiguration controller, performing neural network-based classification of faults occurring during remote modification of intelligence of mobile robots for space application, will increase the efficiency of the diagnostic support of mobile robots, since it is capable of changing diagnostic models at low hardware levels, thus providing great flexibility of diagnostic algorithms and their adaptation to complex types of malfunctions and failures that can occur in mobile robots for space application.

The developed software-algorithmic support of the reconfiguration controller provides a reliable solution to the problem of repairing the faults, occurring in the process of remote modification of intelligence of mobile robots for space application, by identifying the type of the occurred fault, the place of its appearance, as well as the fact that more than one fault occurred.

5. ACKNOWLEDGEMENTS

This work was carried out with the financial support of the Ministry of Education and Science of the Russian Federation in the framework of Agreement No. 14.574.21.0102, 08.09.2014, the unique identifier is RFMEF157414X0102.

6. REFERENCES

1. Moubarak, P., & Ben-Tzwi, P. (2012). Modular and Reconfigurable Mobile Robotics. *Robotics and Autonomous Systems, 60*(12), 1648-1663.

2. Merz, T., Rudol, P., & Wzorek, M. (2006). Control System Framework for Autonomous Robots Based on Extended State Machines. In *International Conference on Autonomic and Autonomous Systems (ICAS'06), Silicon Valley, California, USA, July 19-21, 2006*. IEEE Computer Society.

3. Aitken, J., Veres, S., & Judge, M. (2014). Adaptation of System Configuration under the Robot Operating System. In *19th IFAC World Congress, Cape Town, South Africa, August 24-29, 2014* (pp. 4484-4492).

4. Hernandez, C., Bermejo-Alonso, J., Lopez, I., & Sanz, R. (2013). Three Patterns for Autonomous Robot Control Architecting. In *The Fifth International Conference on Pervasive Patterns and Applications PATTERNS* (pp. 44-51). IARIA.

5. Asha, R., Bowrna, P., & Mhaboobkhan, F. (2013). Implementation of Feed forward Neural network Using Layer Multiplexing for Effective Resource Utilization in FPGA. *International Journal of Research in Engineering and Advanced Technology, 1*(2).
6. Karthikeyan, A., & Rajeswaran, N. (2012). Design and Implementation of Multiple Fault Diagnosis on VLSI Circuits Using Artificial Neural Networks. *International Journal of Advances in Engineering & Technology, 3*(2), 685-695.
7. Valdes, A., Khorasani, K., & Liying, Ma. (2009). Dynamic Neural Network-Based Fault Detection and Isolation. In W. Yu., H. He, & N. Zhang (Eds.), *Advances in Neural Networks – ISNN 2009* (Vol. 5553, Part 3, pp. 780-793). Berlin, Heidelberg: Springer-Verlag.
8. Giger, G., Kandemir, M., & Dzielski, J. (2008). Graphical Mission Specification and Partitioning for Unmanned Underwater Vehicles. *Journal of Software (JSW), 3*(7), 42-54.
9. Sutton, R.S., & Barto, A.G. (1998). *Reinforcement Learning: An Introduction*. MIT Press.
10. Gokhale, M., Graham, P., Wirthlin, M., Johnson, D.E., & Rollins, N. (2006). Dynamic Reconfiguration for Management of Radiation-Induced Faults in FPGAs. *International Journal of Embedded Systems (IJES), 2*(1/2).
11. Bolchini, C., Miele, A., & Santambrogio, M.D. (2007). TMR and Partial Dynamic Reconfiguration to Mitigate SEU faults in FPGAs. In *DFT '07. 22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems* (pp. 87-95).
12. Ivchenko, V., Krug, P., Morozova, T., Ostroukh, A., & Pavelyev, S. (2014). The Remotely Reconfigurable Intelligence of the Space-Based Mobile Robot. *Journal of Engineering and Applied Sciences, 9*, 389-395.
13. Al-Jumach, A.A., & Arslan, T. (1998). Artificial Neural Network Based Multiple Fault Diagnosis in Digital Circuits. In *1998 IEEE Conference Paper* (Vol. 2, pp. 304-307).
14. Al-Jumach, A.A., Alkadim, H., & Arslan, T. (1997). Functional Fault Diagnosis of Mixed Analogue/Digital Circuit Boards Using Artificial Neural Networks. In *Proceedings of the IASTED International Conference, Control '97, Mexico* (pp. 243-247).
15. Kagle, B., Murphy, J., Koos, L., & Reeder, J. (1991). Multi-Fault Diagnosis of Electronic Circuit Boards Using Neural Networks. In *IJCNN International Joint Conference on Neural Networks, San Diego, CA, USA, June 17-21, 1990* (Vol. 2, pp. 197-202).
16. Kagle, B., & Murphy, J. (1990). Neural Network Diagnosis of Multiple Fault Conditions in Electronic Circuit Boards. In *Proceedings of the 1st Workshop Neural Networks Academy/Industrial/NASA/Defence, Auburn University, June 17-21, 1990*.
17. Manikandan, V., & Devarajan, N. (2007). SBT Approach towards Analog Electronic Circuit Fault Diagnosis. *Active and Passive Electronic Components, 2007*, Article ID 59856.
18. Aminian, M., & Aminian, F. (2000). Neural-Network Based Analog Circuit Fault Diagnosis Using Wavelet Transform as Preprocessor. *IEEE Transactions on Circuits and Systems II, 47*(2), 151-156.
19. Wiiltnan, W. (1991). Signature Analysis: A General Neural Network Application in Process Monitoring. In *SME Conf. Neural Network Applications for Manufacturing Product / Process Control, Novi, Mich, USA, April 1991*.
20. Naidu, S.R., Zafiriou, E., & McAvoy, T.J. (1990). Use of Neural Networks for Sensor Failure Detection in a Control System. *IEEE Control Systems Magazine, 10*(3), 49-55.
21. Sorsa, T., & Koivo, H.N. (1993). Application of Artificial Neural Networks in Process Fault Diagnosis. *Automatica, 29*(4), 843-849.
22. Dubey, P., & Kumar, M. (2010). An Advanced Automatic Electronic Diagnosis System. *S-JPSET, 1*(1), 27-31.
23. Ruan, S., & Zhou, Y. (2009). Dynamic Multiple-Fault Diagnosis with Imperfect Tests. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 39*(6), 1224-1236.
24. Satish, K. (2007). *Neural Network: A Classroom Approach* (2nd ed.). Tata McGraw-Hill Education.
25. Andrejevic, M., & Litovski, V. (2004). ANN Application in Electronic Diagnosis – Preliminary Results. In *Proc. 24th International Conference on Microelectronics (MIEL 2004), May 16-19, 2004* (Vol. 2, pp. 597-600).
26. Stopjakova, V., Malosek, P., & Nagy, V. (2006). Neural Network-Based Defect Detection in Analog and Mixed IC Using Digital Signal Preprocessing. *J. Electrical Engineering, 57*(5), 249-257.
27. Sridhar, K.P., Vignesh, B., Saravanan, S., Lavanya, M., & Vaithiyanathan, V. (2014). Design and Implementation of Neural Network Based circuits for VLSI Testing. *World Applied Sciences Journal, 29*, 113-117.