UDC 519.22/.25 (045)
DOI:10.18372/1990-5548.58.13520

A. A. Zelenkov

IMITATION MODELING OF THE RECOVERY PROCESS OF THE ON-BOARD FAULT-TOLERANCE COMPUTER SYSTEM

Educational & Research institute of Information and diagnostic Systems, National Aviation University, Kyiv, Ukraine
E-mail: elte.chair @ gmail.com

Abstract—*The possibility of imitation modeling of automatic recovery of a computer fault-tolerance system, whose elements have additional hardware and software redundancy in case successive failures, is considered on the base of a directed probabilistic graph whose tops correspond to possible states of the system, and the arcs between them determine the probabilities of transitions from one state to another, the arc length determines the random time of automatic recover, statistical characteristics of the recovery process are determined on the base of passage of the routes along the probabilistic graph from initial top to the final one.*

Index Terms—Fault-tolerance system; failure; recovery time; probability of recovery; directed graph; imitation modeling; automatic recovery; hardware and program reserve; transition time; failure localization.

I. INTRODUCTION

Fault-tolerance structures of avionics must provide a high level of automation at all stages of flight: at the route, automatic approach and landing on a III category ICAO, automatic control by a run after landing [1]. Fault-tolerance computer system processes information on board of the aircraft in real time and with a high degree of probability should guarantee that the failure of the system (or functional module) will be detected and localized with the subsequent recovery of the database and the computing process.

The quality of the such system can be estimated by the probability with which it guarantees automatic recovery in case of failure of individual elements, average recovery time of the functional modules and the system as a whole, by the number of failures of elements, which still maintain the functionality of the system unit (failure of the *i*-th multiplicity), which is equivalent to the number of operational states of the system in case successive failures. The number of such failures determines the survivability of the fault-tolerance system.

In accordance with the requirements for fault-tolerance structures of avionics it is necessary that all functions continue to be performed for 250 hours after the first failure with a confidence level of 0.99.

II. PROBLEM STATEMENT

In accordance with the strategy for the development and implementation of perspective fault-tolerance system of avionics it should be provided possible to delay the maintenance procedure until the aircraft returns to the main base, which in turn will allow for the implementation of the planned maintenance intervals. It is evident that such concept can be achieved only by keeping the failure in a given time interval after its detection and localization. Repair of the system for the period of its short-term operation is not expected.

Thus, although redundancy of system components is necessary, however, the main emphasis is placed on the wider use of keeping failure techniques, which allows other functional elements to continue functioning in the presence of a fault. The basic properties and characteristics of fault-tolerance systems are considered in [2].

In general any fault-tolerance system may be described by the state graph. Then it is obvious that at the design stage of such systems it is necessary to evaluate the reliability characteristics of the systems at a certain time interval of possible operation.

III. PROBLEM SOLUTION

For quantitative evaluation of functioning fault-tolerance computer system mathematical model in the form of a directed (oriented) graph of the process of automatic recovery of a system that has an appropriate level of redundancy can be used.

Such representation of the system functioning process allows to determine the following characteristics by means of imitation modeling.

The main parameters of fault tolerance are:

– recovery time during which the performed function will recover from failure,

– system response time to failure detection and performing the necessary reconfiguration.

In case of failure of the backup module, the system can go into state of failure that should be

reflected by the corresponding transition line (arc) in the state graph.

In general case, the graph may show the recovery process of a separate functional module, which has additional hardware and/or software to back up its elements, and the entire system.

It is evident that the transition time from one state to another is a random variable. After each failure of any element, the operability of the fault-tolerance system is recovered (that is a certain route is implemented) in a certain random time τ_{rec} with the distribution function $F(\tau_{rec})$. There is a critical (permissible) time τ_{per} to recover system operation in real time, the excess of which means its failure.

The probabilities of branching along the arcs of the graph p_{ij} can be determined in accordance with the proportion of equipment controlled by means of hardware and software control. It is possible to set the scatter of these probabilities with some given variance, but so that $p_{ik} + p_{il} = 1$.

The block diagram of imitation modeling for the recovery process of a certain functional module is shown in Fig. 1.

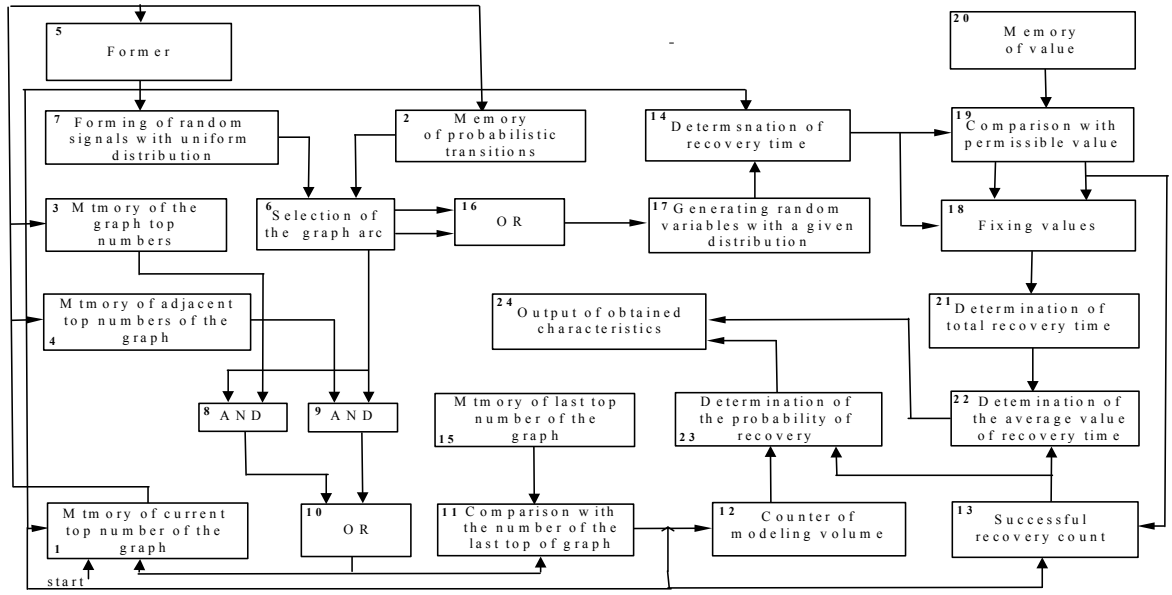

Fig. 1. Block diagram of imitation modeling for the recovery process

Such reliability indicators are determined as the average time and the probability of recovery in a given time. The maximum permissible recovery time for τ_{per} is known.

The probabilities of transitions from those tops of the graph that have two outputs are also known (imitation of a logical element). Obviously, the only output of the operating element (state with a single output) is activated with a probability of one.

The length of the arc of the graph is determined by the time between two corresponding events (states) in the computer system. This time is a random value and when modeling is realized in accordance with a given distribution law.

Statistical characteristics are determined as a result of multiple passages of those or other routes of a probability graph. The number of such routes determines the volume of modeling *M*. If the route length is greater than τ_{per}, then this realization of route is excluded when calculating the average time and the probability of recovery:

$$P_{rec} = \frac{N}{M}, \qquad \overline{\tau}_{rec} = \frac{\sum_{i=1}^{N} \overline{\tau}_{rec,\,i}}{N},$$

where *N* is the number of successful route realizations for which the recovery time is less than the permissible value.

Blocks 1–11 define various realizations of routes on a probability graph. The accumulative adder 14 determines the recovery time of the *i*th route, and adder 21 determines the total time of successful recoveries. The reversible counter 13 fixes the number of successful recoveries *N*, and counter 12 fixes the volume of modeling *M*. The generator 7 forms a continuous random value with a uniform distribution law on the interval [0; 1]. The generator 17 forms a continuous random value with a given distribution law, which determines the random time between neighboring states (tops).

In block 2 the values of the transition probabilities are stored in accordance with the sequential numbers of the tops of the graph. For

example, in the first cell that corresponds to the top Z_1 (top number 1) the smallest transition probability value (p_{12} or p_{13}) is written, and the value in the second memory cell is either p_{24} or p_{25}, one is written in the fourth cell (corresponding to the fourth top) and etc., Fig. 2.

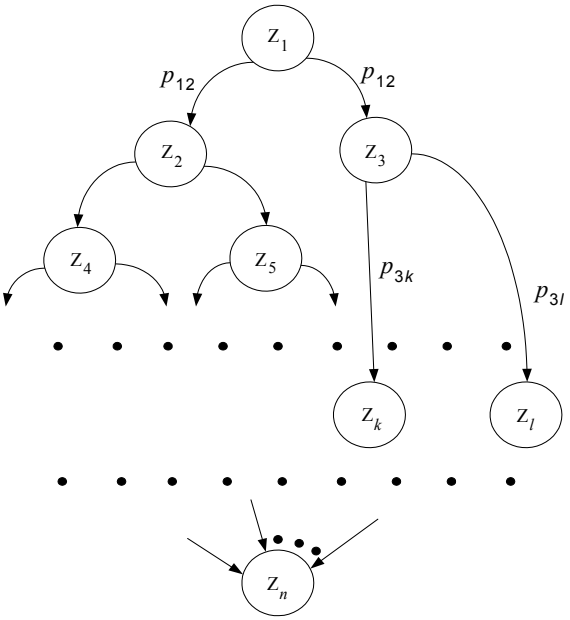

Fig. 2. The directed graph of recovery process

In block 3, the smallest top number of the graph is written, which is connected with the current top, and in block 4 the second top number is written, which is connected with the current top (in the absence of such top, zero is written.

Let, for example, at the output of block 3 the code of the current top Z_3 is set, and $p_{3,k} < p_{3,l}$. Then at the output of block 2 will be set the value $p_{3,k}$, and at the output of block 3 will be set number k, at the output of block 4 will be set the number l, i.e. the numbers of the tops which are connected with the current top with number Z_3.

Thus, in three memory blocks a specific probabilistic graph is written. For another graph memory blocks are overwritten.

When the signal "start" is applied at the control input of block 1, the first top code Z_1 of graph appears at its outputs, which sets on the outputs of memory blocks respectively: the probability value p_{12} (under the condition that $p_{12} < p_{13}$), the code of the number 2 (block3) and the code of the number 3 (block 4). Besides, the signal from the output of register 1 using the former 5 launches the generator 7, at the output of which a random value r_i appears from the interval [0; 1]. If $r_i < p_{12}$, then this means that an arc was realized in the graph, which corresponds to the transition probability p_{12} (in another case the arc p_{13} is realized). Then for the case $r_i < p_{12}$ a signal appears on the first output of the comparator 6, which allows the code of the second top number to pass through the coincidence scheme 8. The signal at the second output of the comparator blocks coincidence scheme 9.

If $r_i > p_{12}$ then the second output of the comparator 6 is activated. Besides, the signal from the corresponding output of block 6 through the element OR 16 is applied to the input of the generator 17, activating at its output a random value that is equal to the length of the arc of the graph between the states Z_1 and Z_2. This value is applied to the information input of accumulating adder 14. Next, the code from the output of block AND 8 through the OR 10 (at the output of block 10 the code of the current graph top is generated while passing the route) is applied to the information input of register 1 and then is transmitted to the address inputs of all memory blocks. Then, at the input of block 2 the code p_{24} ($p_{24} < p_{25}$) is set, at the outputs of blocks 3 and 4 the top codes Z_4 and Z_5 are set.

If $r_i > p_{24}$, then the code of top Z_5 is transmitted to the output of block 10 and then this top becomes the current top, etc.

After passing the route, the code of the last top Zn is set at the output of block 10 and the code corresponding to the recovery time for the implemented route is created at the output of the adder 14. If its value does not exceed the allowable value, which is set by block 20, then a signal appears at the output of block 19 which rewrites the contents of adder 14 using the register 18 into the accumulating adder 21. Otherwise, the signal appears on the other output of block 19 and sets the register to the zero state and subtracts one from the content of the reversible counter 13. Besides, the current top number at the output of block 10 is applied to the input of comparison block 11 where it is compared with the code of the last top Z_n, which comes from block 15. If the codes match, then a signal appears at the output of comparator 11 which adds one to counter 12 and the reversible counter 13. Next this signal sets the adder 14 in the zero state and the code of the first top Z_1 is written in register 1. Further process repeats.

Thus, at the output of accumulative adder 21 the current sum of the recovery time of all successful routes is set, so that at the outputs of blocks 22 and 23 the current estimates of the average time and probability of recovery are obtained, which are reflected in the indicator.

The considered work algorithm can be realized both in hardware (like in Fig. 1) and in software.

IV. CONCLUSIONS

During the design and development stages of a fault = tolerance system many parameters of the recovery process are unknown. Their estimates can be obtained after carrying out imitation modeling of the system represented by directed graph of states, while it is possible to vary the types and parameters of the distribution laws of time intervals in accordance with physical considerations.

Analysis the recovery process of the entire system, represented by a superposition of individual functional modules, each of which has its own means of recovery, can be carried out by decomposition. Besides a comparative estimation of the quality of the recovery process can be carried out for each module in order to determine their potential possibilities and the respective redistribution of hardware and software means of control. Considered principle of imitation modeling can be used for any complex computer system.

REFERENCES

[1] A. A. Zelenkov and V. M. Sineglazov, *On-board automatic control systems. Accuracy estimation of flight test results*. Kyiv: NAU, 2009, 264 p. (in Russian).

[2] K. A. Yiudu and S. A. Krivoschekov, *Mathematical models of fault-tolerance computer systems*. Moscow: MAI, 1989, 144 p. (in Russian).

Received November 6, 2018

Zelenkov Alexander. Candidate of Science (Engineering). Professor.
Computerized Electrical Systems and Technologies Department, National Aviation University, Kyiv, Ukraine.
Education: Kyiv Civil Aviation Engineers Institute, Kyiv, Ukraine, (1968).
Research area: Estimation of the accuracy and reliability of on-board automatic control systems.
Publication: 239.
E-mail: elte.chair@gmail.com

О. А. Зеленков. Імітаційне моделювання процесу відновлення бортової відмовостійкої обчислювальної системи
Розглянуто можливість імітаційного моделювання процесу автоматичного відновлення відмовостійкої обчислювальної системи, елементи якої мають додаткове апаратне та програмне резервування при послідовних відмовах, на основі орієнтованого ймовірнісного графа, вершини якого відповідають можливим станам системи, а дуги між ними є ймовірностями переходів від одного стану до іншого, при цьому довжина дуги визначає випадковий час автоматичного відновлення; статистичні характеристики процесу відновлення визначаються на основі проходжень маршрутів уздовж ймовірнісного графа з початкової вершини до кінцевої.
Ключові слова: відмовостійка система; відмова; час відновлення; ймовірність відновлення; орієнтований граф; імітаційне моделювання; автоматичне відновлення; апаратне і програмне резервування; час переходу; локалізація відмови.

Зеленков Олександр Аврамович. Кандидат технічних наук. Професор.
Кафедра комп'ютеризованих електротехнічних систем та технологій, Національний авіаційний університет, Київ, Україна.
Освіта: Київський інститут інженерів цивільної авіації, Київ, Україна, (1968).
Напрям наукової діяльності: Оцінка точності і надійності бортових автоматичних систем управління.
Кількість публікацій: 239.
E-mail: elte.chair@gmail.com

А. А. Зеленков. Имитационное моделирование процесса восстановления бортовой отказоустойчивой вычислительной системы
Рассмотрена возможность имитационного моделирования процесса автоматического восстановления отказоустойчивой вычислительной системы, элементы которой имеют дополнительное аппаратное и программное резервирование при последовательных отказах, на основе ориентированного вероятностного графа, вершины которого соответствуют возможным состояниям системы, а дуги между ними определяют вероятности переходов от одного состояния к другому, при этом длина дуги определяет случайное время автоматического восстановления; статистические характеристики процесса восстановления определяются на основе прохождений маршрутов вдоль вероятностного графа из начальной вершины в конечную.
Ключевые слова: отказоустойчивая система; отказ; время восстановления; вероятность восстановления; ориентированный граф; имитационное моделирование; автоматическое восстановление; аппаратное и программное резервирование; время перехода; локализация отказа.

Зеленков Александр Аврамович. Кандидат технических наук. Профессор.
Кафедра компьютеризированных электротехнических систем и технологий, Национальный авиационный университет, Киев, Украина.
Образование: Киевский институт инженеров гражданской авиации, Киев, Украина, (1968).
Направление научной деятельности: Оценка точности и надежности бортовых автоматических систем управления.
Количество публикаций: 239.
E-mail: elte.chair@gmail.com