Methodology of Defining the Accident Rate Function for Fault Tolerant System with High Responsibility Purpose

Leonid Ozirkovskyy, Bohdan Volochiy, Mykhailo Zmysnyi, Andriy Maschak

Lviv Polytechnic National University, 12 S. Bandera street, Lviv, Ukraine, 79000
l.ozirkovsky@gmail.com, bvolochiy@ukr.net, zmysnyim@gmail.com, himakus@gmail.com

Abstract. In this paper we propose a new term - accident rate function. Such term gave a possibility to provide the quantative assessment for operational safety in the fault tolerant systems with high responsiblity purposes.
Moreover, we propose a binary structural automata model. Using the proposed binary structural automata model in the ASNA software, we provide a possibility to build models of the fault tolerant systems in the form of a graph of states and transitions, in an automatic way. Obtained graph of states and transitions is used to define the accident rate function.

The authenticity of the emergency rate function is confirmed by the coincidence of two calculated values. One value is obtained based on accident rate function at determined time interval and the other value is the probability of minimal cut sets obtained based on fault tree at a similar time interval. Using the ASNA software to get the accident rate function and the usage of new methodology of forming the accident rate function from the subarray of non-functioning states makes the process of obtaining the results in an automatic way. As a result, the proposed approach gives a possibility to perform multivariant analysis of functional safety for the systems with high responsibility purpose.

Keywords: Safety Analysis, Reliability Model, Fault-Tolerant System.

1 Introduction

If the high quality of critical system functioning is required, then the required level of its reliability must be ensured. There are such critical systems as a control system for transport vehicles (aviation, railway, marine transport), a control system with power objects (nuclear, thermal, hydroenergetics), military-oriented systems, medical systems. For such systems, the most determined property is the pre-defined level of functional safety. Functional safety is the property of the system and it determines whether the system or the system submodule fails, and if it does, then the system switches to the state in which there are not any harmful consequences for humans or environment or dependant systems/submodules [1 - 3].

An assesment of functional safety of exploitation of the critical systems is carried

out using the following analyses - Failure Mode, Effects and Criticality Analysis (FMECA), Fault Tree Analysis (FTA) and Even Tree Analysis (ETA). As a result, we obtain the values of the exploitation risk. The proper values are set according to international standards [1, 2, 20]. In case of exceeding maximum legitimate value, it is necessary to provide some corrective actions to decrease it.

For the functional safety, the main value of risk exploitation is a probability of the minimum cut sets (MCS). The minimum cut set is the smallest combination of events called “element failed”, which results in failure of the whole system. If one of the “element failed” events is removed from MCS, the system failure is impossible [4 - 6]. The analysis of MCS gives a possibility to present the most vulnerable elements of the system. As a result, the functional safety of the whole system can increase if the safety of only the most critical (vulnerable) elements of the system expands.

Based on the literature review of functional safety assesment, we may assume that a basic methodology (as an instrument) is a Fault Tree Analysis. The improvements of MCS determination methodology based on FTA are shown in the papers [9, 10, 11, 12, 14]. Methods of time decreasing for MCS determination using FTA, and the methods of calculation probabilities are shown in the publications [7, 8, 13, 15]. It can be seen that in FTA the tree building is the largest and the most time-consuming operation. Therefore, this operation requires a considerable skill from a designer.

The main use of FTA is acceptable only when you need to analyse the functional safety of complex systems only once. But FTA usage is not applicable on the design stage, when it is necessary to execute the functional safety assesment for each proposed variant of the system. The question is about the synthesis of the system with required level of functional safety.

The functional safety assesment, based on the Markov model for the complex system, is shown in the papers [16, 17]. In a monography [16], the approach is developed to obtain the MCS from the graph of the states and transitions. This approach is intended to be used for the analysis of the systems where its model has a large dimension (count with the number of states more than a million). The method of simplification of the graph of the states and transitions was used as a basis for the proposed approach. The simplification has a rule to unite the “similar” states. In the article [17], the approach of safety assessment using the Markov model is presented. As the safety index, the Mean Time Until Failure is used.

Development of the fault tolerant system of the responsibility purpose with the pre-defined level of functional safety on the design stage foresees tackling the task of synthesis of functionality and reliability. Such tasks can be solved with the analysis of many expedient variants of the system. To solve such task, the designer must have a methodology based on which he can determine the functional safety for lots of system variants in a limited interval of time.

2 Accident rate function for fault tolerant system

For the safety assessment of fault-tolerant systems, the MCS are used. The MCS are presented as logical functions [18, 19]. Note that obtained MCS using the fault tree

are point-based and these MCS represent the specific value for the time of operation. For the designer of the fault tolerant system, it is useful to have the dependence of the MCS occurrence probability value and the change of the time of operation. To obtain such a characteristic, many fault trees must be developed, since the fault tree is constructed for given operating time.

Using the model of the fault tolerant system in the form of the graph of the states and transitions opens up the possibility to determine the MCS occurence probability for any value of the time of operation [22].

The time spent on developing complex system model in the form of the fault tree is comperative to the development of the graph of states and transitions [22]. However, if the building of the state graph is automated [3], then it is possible to determine the MCS occurrence probability, depending (as a function) on the time of operation of the system which is under investigation. We propose to name this function as an "accident rate" function.

Accident rate function (ARF) is dependence on the time of operation (observation), and the probability of the system in the failure state, which leads to an accident. For example, the value of the accident rate is defined as the sum of the probabilities of staying in safe non-functioning states, critical and/or catastrophic states. The transitions between these states show the trajectory of the transition (evolution) of the system from the insignificant failure to failure. Moreover, the less transitions from a failure safe state to a catastrophic one occur, the lower level of the functional safety system has. And therefore, there are fewer opportunities to avoid an accident.

According to the results of the provided research for the accident rate function, the following properties are established:

1) For a particular system, the number of accident rate functions Q_A (t) equals to the minimum cut sets (MCS) for an accident.

2) The value of the accident rate function at a specific moment of time is equal to the probability of the appearance of the minimum cut set, which is obtained using the fault tree for the same time period.

3) The probability of occurrence of an accident rate situation Q_{AC} (t) at a given interval of time is determined by the following formula:

$$Q_{AC}(t) = 1 - \prod_{i=1}^{k} [1 - Q_{Ai}(t)] \quad (1)$$

where $Q_{Ai}(t)$ – *i-accident rate function*,

k – a number of accident rate functions.

The methodology of determining the accident rate function is shown in the section below.

3 Creation of the mask to select the non-functioning states which form the accident rate function

To get the formula of the accident rate function, we need to define the space of non-functioning states. These non-functioning states cause the accident situation. Because some non-functioning states could be in the different ARF, it is required to

have the means to define their identification. As such means, we propose to use the mask of the accident rate situation.

The accident rate mask is a logical function, formed from the components of the vector states, the transition into a non-functioning state which is necessary and there is a sufficient condition to make the accident situation occur. The accident rate mask is obtained from "Condition of fault of the fault tolerant sytem fail" by minimizing it under the rules of algebra of logic.

The accident rate mask has the following properties:

If the logical expression, which describes the accident situation, consists of components of the vector states (VS), united only by the operator "AND", then for the research object there is one accidental function:

$$(Vg=0)\wedge(Vh=0)\wedge\ldots\wedge(Vk=0)$$

If the logical expression, which describes the accident situation, consists of groups of components united by the operator "OR", and in each of the groups, the VS components are combined only by the operator "AND", then the z-functions of the accident rate are inherent in the object of the investigation:

$$((Vm=0)\wedge(Vn=0)\wedge\ldots\wedge(Vq=0))\vee\ldots..\vee((Vs=0)\wedge(Vt=0)\wedge\ldots\wedge(Vy=0))$$

For instance, if as a result of minimization for the "Condition of fault the fault tolerant sytem fail" the following function was obtained and it consists of three groups of the MCS components which are combined by the OR logical operator:

$$((V1=0)\wedge(V2=0)\wedge(V4=0))\vee((V2=0)\wedge(V5=0))\vee((V1=0)\wedge(V5=0)),$$

then in this case, there are three accident rate functions. The first function of the accident rate is formed by non-functioning states of the system in which the 1st, the 2nd and the 4th modules fail. The second ARF forms non-functioning states of the system, in which the 2nd and the 5th modules fail, and the third one - the non-functioning states of the system in which the 1st and the 5th modules are non-functioning

Based on the obtained masks, using the special algorithm, which is given below, the ARF is formed.

4 Algorithm to form the accident rate functions from the sub Space of Non-Functioning States

The algorithm of the ARF formation consists of two stages. At the first stage, the groups of all the states are determined based on the mask of an accident rate which correspond to a specific ARF. At the second stage - the expressions are formed to calculate the quantitative value of ARF, from the selected states.

4.1 Stage I: Defining the groups of the states which correspond to each accident rate function

All the states are selected, in which the VS components correspond to the mask of the accident rate equal to zero. If the mask of the accident rate has several components integrated by the logical OR operator, then there will be ARFs, and there will be the selected group of states for each of them.

The input data for the algorithm is a set of non-functioning states, that are obtained using a binary structural automata model (SAM).

When developing the algorithm for automated determination of ARF , the following assumptions were adopted:

• at least one ARF is inherent to the system;

• a catastrophic state (CS) is the state of the fault system with high responsibility purpose (FSHRP), which creates an accident rate on the object of its (use);

• the accident rate function of the system is determined by a set of states, in which the system enters the path to the fault (accident). If at least one VS component, which is zero, in all these states has given the value of one (to be put into functioning condition), then the creation of an accidental situation will not be due to the FSHRP.

For a compact (algorithmic) description of the developed method, the following abbreviations are used:

n – a pointer to the ordinal number of the ARF.

i, j - indicators of the ordinal number of the VS components.

CSC - a counter for the number of components in an accident rate mask (the number of expressions separated by operators OR).

ECC – an external cycle counter.

ICC - an internal cycle counter.

CNCVS - a constant number of components of the vector state.

CNC - a counter of the number of VS components in the accident rate mask.

ZCn - a zero counter; this counter for ARF, with ordinal number n the number of VS components which value equals to zero.

ARFC - ARF counter.

ARMC - the mask of an accident rate component.

AARF - an array of ARFs.

SE - a sign of equality.

SARF - a sign of ARF.

Vcn [i] - the value of the i-th state vector of the mask component of the accident situation with the ordinal number n.

To find ARF, it is necessary to sort the obtained array of non-functioning system states on the basis of the smallest number of events which led to the accident rate of the system with the minimum number of VS components equal to zero. They are non-functioning states, in which the transition was made directly from functioning state. As a rule, they are non-functioning safe states. On the basis of the sorted array of non-functioning system states in accordance with the mask component of an accident rate, there are the system states which serve to form the specific ARF. As a result, the array of ARF is obtained.

The first step of this phase is to create a matrix which will consist of three columns – in the first column the ordinal number of the mask of accident rate component is written – N, in the second one – VS component is written using comma which

corresponds to the first component of accident rate mask and its value, in the third one – the value of zero counter – especially the number of zeros of VS component in their respective VS (ZC).

ARF sorting procedure

The sorting procedure is performed in two embedded cycles – external and internal ones – by comparing two adjacent components of the accident rate mask, and involves the execution of the following steps.

The input data:

ECC - an external cycle counter which is assigned with the value of a number of components of the accident rate mask.

ICC = (ECC - 1)

n – a pointer to the ordinal number of mask component of an accident rate.

(n+1) – a pointer to the next ordinal number of mask component of an accident rate (of an accident rate)

Step 1. **n=1** is assigned to the pointer to the ordinal number of mask component of an accident rate system; a pointer to the next ordinal number receives the value **(n+1)**=**2**, the unit is subtracted from the counter of an external cycle **ECC, -ECC = ECC-1**; Then you should check the condition whether **ECC** equals to zero:

If ECC = 0, it means that the sorting procedure of the mask components of an accident rate is considered to be **completed**. As a result of such procedure, the matrix of mask components of an accident rate is obtained where they are sorted according to the number of VS components which value equals to zero. So, at the beginning the states with the smallest number of VS components which value equals to zero will be introduced.

If **ECC > 0,** it means that the sorting procedure of the mask components of an accident rate continues and it is necessary to proceed to step 2.

Step 2. At this stage the counter of $\mathbf{ZC}_n$ zeros with the number n is compared to the counter of $\mathbf{ZC}_{(n+1)}$ zeros with the number **(n+1).**

If the value of the counter of $\mathbf{ZC}_n$ zeros with the ordinal number n is bigger than the value of the counter of $\mathbf{ZC}_{(n+1)}$ MCS with the ordinal number **(n+1)**, then these VS must be swapped; then it is necessary to reduce the counter of an internal cycle **ICC = ICC - 1** and proceed to step 3.

Note. In the matrix the mask component of an accident rate of ordinal numbers should not be changed, only mask components should be swapped which means to swap VS.

If the value of the zero counter in the mask component of an accident rate $\mathbf{ZC}_n$ with the ordinal number n is smaller or equals to the value of the zero counter of $\mathbf{ZC}_{(n+1)}$ component with the ordinal number **(n+1)**, which means that these components are not swapped; Then it is necessary to reduce the counter of internal cycle **ICC = ICC-1** and proceed to step 3.

Step 3. At this stage it is necessary to increase the pointer to the ordinal number of

the mask component of an accident rate adding one **n = n + 1** and the next pointer to the ordinal mask component **(n+1) = (n+1) + 1.** It is also necessary to check whether the value of counter of internal cycles does not equal to zero:

If **ICC > 0,** it means that not all adjacent mask components of an accident rate were compared, so it is necessary to proceed to step 2;

If **ICC = 0,** it means that all adjacent mask components of an accident rate were compared, so it is necessary to proceed to step 1;

As a result of moving mask components of an accident rate in the matrix, the sorted matrix of these components is obtained. The sorting was based on the value of the number of zeros in MCS. In the first line of the obtained matrix, there will be VS with the smallest number of zeros which corresponds to the mask component of an accident rate. Then there will be the mask component with the same or bigger number of VS components which equals to zero and until all the states are selected, which value of VS components corresponds to the accident rate mask.

Method of determining the accident rate functions

Method of determining ARF uses the following procedures: to find ARF and to compare the mask component of an accident rate. The process of finding ARF takes place in several embedded cycles – the general cycle of finding ARF and the internal cycles of the comparing procedure of mask components of an accident rate.

The input data:

CNC – the value of a number of mask components of an accident rate is assigned;

ARFC – zero is assigned to ARF counter **ARFC = 0**;

j- pointers to the serial VS component;

n- a pointer to the ordinal number of CC_n;

SE - a sign of equality.

SARF - a counter of ARF sign.

Step 1 . The pointer to the ordinal number of mask component of an accident rate obtains the value of the number of components in a mask minus a unit **– n = CNCVS** and it is necessary to proceed to step 2.

Step 2. The pointer to the ordinal VS component obtains the first value and the counter of ARF sign obtains the value of the pointer to the ordinal number CC – **j=1**; **SAFR=n**. It is necessary to proceed to step 3.

Step 3. At this stage it is necessary to use the comparison procedure of mask comparison of an accident rate (MCAR). The input data will be the following ones: n- the pointer to the ordinal number ON and j – the pointer to the ordinal VS component. After MCAR execution it is necessary to check the sign of equality SE:

If after the MCAR execution the sign of equality will equal to zero **SE=0**, it means that the counter of ARF sign should be reduced by one **SAFR = SAFR - 1**, and proceed to step 4.

If after the MCAR execution the sign of equality will not equal to zero **SE=0**, it is necessary to proceed to step 4.

Having increased the pointer to the ordinal VS component, the next component is selected VS – j = j + 1; However, it is necessary to check whether such VS component exists so that such condition is to check:

If **j > n,** it is necessary to proceed to step 5.

If **j** ≤ n**,** it is necessary to proceed to step 3.

Step 5. This step checks whether the mask component of an accident rate with the ordinal number n is ARF. It is done by checking the counter of MCS sign which equals to zero or not.

If **SAFR** = 0, it means that MCS with the ordinal number n is the AFR system. The mask component with the ordinal number **n** should be written in an array of accident rate functions **AARF**, and increase the counter MCS by a unit **ARFC = ARFC + 1**; then proceed to step 6.

If **SAFR** > 0, it is necessary to proceed to step 6.

Step 6. **n = n - 1**;

If **n > 0,** it is necessary to go to step 2.

If n=0, it means that all MCS are checked and all procedures to find ARF were completed and all states of a graph which form the specific ARF were found. The procedure to find ARF is completed.

The procedure of comparing the system states

The input data:

CNCVS – is assigned to the value of the number of VS components;

$\mathbf{ZC_n}$ - is assigned to the number of zeros of VS components in a state with the ordinal number n.

$\mathbf{VS_n\ [i]}$ – the value of **i** of VS component that corresponds to the mask component of an accident rate with the ordinal number **n.**

$\mathbf{VS_j\ [i]}$ – the value of **i** component of the vector state that corresponds to the mask component of an accident rate with the ordinal number **j.**

i = l;

The input data is also the obtained data from the procedure of finding ARF especially **n** and **j.**

Step 1. At this stage the relevant VS components are compared to the mask components of an accident rate.

If $\mathbf{BC_n[i]}$ component equals to zero $\mathbf{(BC_n[i] = 0)}$ and $\mathbf{BC_j[i]}$ component also equals to zero $\mathbf{(BC_j[i] = 0),}$ it means that $\mathbf{ZC_n = ZC_n}$ -1; **i = i + 1**; then it is necessary to proceed to step 2.

If any of the above mentioned conditions is not fulfilled, it means that **i = i + 1**; the it is necessary to proceed to step 2.

Step 2. At this stage the current number of VS component is checked whether it exceeds the total number of VS components of an accident rate in the mask.

If i ≤ **VSC,** then go back to step 1.

If i > **VSC,** then proceed to step 3.

Step 3. At this stage the certain value is attributed to the **SE** comparison sign.

If $\mathbf{ZC_n = 0}$, so 1 is assigned to SE.

IF $\mathbf{ZC_n} > 0$, so 0 is assigned to SE.

At this stage the comparison procedure of the mask components of an accident rate is completed.

As a result of such procedure, there is the return value of the equality sign **SE** to the procedure which triggered it.

4.2 Stage II: Algorithm for forming expressions for the accident rate functions

At the stage II, it is necessary to create the matrix which consists of four columns – in the first column the ordinal ARF number is written – N, in the second one – VS component and its value is written, in the third one – the numbers of states are written that form the specific ARF.

The input data:

ARF array obtained at the Stage 1.

An array of all system states (functioning and non-functioning).

ZC – the counter of ARF amount that is recorded in the **AARF** array**.**

CNCVS – constant total number of the number of system states.

Step 1. j is assigned a unit to the pointer of the ordinal ARF number. It means that the first ARF is selected from the array of accident rate functions – j=1.

Step 2. n is assigned a unit to the pointer of the ordinal number of a matrix component of an accident rate MCAR. It means that the first MCS is selected from the array of all system states – **n=1**.

Step 3. Then it is necessary to use the comparison procedure of matrix components of an accident rate MCAR where **n** and **j** are the initial data.

If after the MCAR procedure, the equality sign will equal to one **SE = 1**, then in the third column in AARF the state number –**n** should be written and proceed to step 4.

If after the MCAR procedure, the equality sign will equal to zero **SE = 0**, then proceed to step 4.

Step 4. The pointer to the ordinal number VSC – **n** is increased by one – **n = n + 1**; It is also necessary to check whether the given pointer has exceeded an array of system states. The check is carried out according to the following condition:

If Якщо **n < CNCVS,** then the given pointer has not exceeded the array of system states, that is why it is necessary to proceed to step 3.

If **n ≥ CNCVS,** then it is necessary to proceed to step 5.

Step 5. The pointer to the ordinal number of matrix component of an accident rate – **j** is increased by one **j = j + 1**; It is also necessary to check whether the given counter has exceeded the ARF array. The check is carried out according to the

following condition:

If **j ≤ ZC,** then the given pointer has not exceeded the array of system states, that is why it is necessary to proceed to step 2.

If **j > ZC,** then the procedure of finding states which posesss the relevant ARF is completed.

As a result of such procedure, the third AARF column is filled in.

The procedure to obtain the ARF expression is to sum the probability values of staying in relevant states, whose numbers were found in the previous procedure, meaning in states which are written in the third column of the relevant ARF in the matrix of ARF array. As a result, the fourth column is filled with the probability values of relevant ARF. Therefore, the expression of an accident rate function equals to the sum of probabilities in those states that correspond to the accident rate mask.

$$Q_{Ai}(t) = \sum_{j=m}^{q} P_j(t) + ... \qquad (2)$$

where $P_j(t)$ – probabilities of MCS stay in a group of non-functioning states m … q, whose value of VS components equals to zero in accordance with **i** accident rate mask. The group of non-functioning states in the simplest case can include all the non-functioning states. There can be several groups of such states for MCS.

For example, if the accident rate mask:

$$(Vg=0)\wedge(Vh=0)\wedge(Vk=0)$$

corresponds to such states 20, 21 … 27 and 32 … 35, then the ARF expression will have such a look (3):

$$Q_A(t) = \sum_{i=20}^{27} P_i(t) + \sum_{i=32}^{35} P_i(t) \qquad (3)$$

5 The methodology validation of determining accident rate functions

The methodology validation of determining accident rate functions is carried out by comparing the results obtained from the universal MCS model in a form of a graph of states and transitions using binary structural automata model (SAM) and the results obtained from the fault tree constructed using the software Reliasoft BlockSim [21] for test MCS.

Test MCS without restoration consists of two different modules connected consecutively. Both modules have hot reserve. The first module, which is less reliable, has two reserve modules, while the second one has 1 module. In case of the main module failure, the backup one connects instead of the main one. Means of control and commutation are considered to be absolutely reliable and fast. Therefore, in the most reliable model the probability of successful control and the probability of successful re-

serve module connection equal to one, and duration of these procedures equals to zero. Reserve modules can malfunction regardless of their main ones.

At the first validation stage, the binary structural automata model (SAM) is constructed. A separate component of vector states corresponds to each module. The initial value of each component of vector states equals to 1, since all modules are functioning at the initial moment of time.

The constructed binary structural automata model (SAM) is appointed for ASNA software tool which constructs a graph of states and transitions on its basis. As a result, the graph of states and transitions is obtained which contains 32 states and 111 transitions. The list of states with the value description of each component of vector states is illustarted in Fig. 1.

According to the methodology, the determination of the array of non-functioning states was carried out. The states 1 – 7, 9 – 15, 17 – 23 are functioning. The states 8, 16, 24-32 are non-functioning. Accident rate functions will be formed out of these states.

In order to form accident rate functions in accordance with the developed methodology, it is necessary to form accident rate masks by minimizing the condition of the catastrophic MCS failure:

((V1=0) AND (V2=0) AND (V3=0)) OR ((V4=0) AND (V5=0))

Since the relatively simple fault-tolerant system is chosen in this example, it means that the condition of catastrophic MCS failure does not require minimization and was immediately written as the disjunction of the conjunctions. As a result, there are two operandas with the disjunction sign and therefore, there are two accident rate functions.

The first accident rate function has the mask - ((V1=0) AND (V2=0) AND (V3=0)), and the second one - ((V4=0) AND (V5=0)).

So, the first accident rate function will be the sum of probabilities in states where V1, V2 and V3 components equal to zero (4). These states are 8, 16, 24 and 32 (look at Fig.1):

$$Q_1(t) = P_8(t) + P_{16}(t) + P_{24}(t) + P_{32}(t) \quad (4)$$

The second accident rate function will be the sum of probabilities in states where V4 and V5 components equal to zero (5). These states are 24-32 (look at Fig.1):

$$Q_2(t) = P_{25}(t) + P_{26}(t) + P_{27}(t) + P_{28}(t) + P_{29}(t) + P_{30}(t) + P_{31}(t) + P_{32}(t) \quad (5)$$

The Kolmogorov–Chapman system of differential equation was compiled using ASNA software and based on the obtained graph of states and transitions, it was solved and the probabilities division in each state was obtained. The obtained division was exported to Excel spreadsheets and accident rate functions $Q_1(t)$, $Q_2(t)$ were constructed which are illustrated on Fig.2.

There is also constructed the dependence of probability of accident rate occurrence on time as the sum of probabilities in all non-functioning states (6):

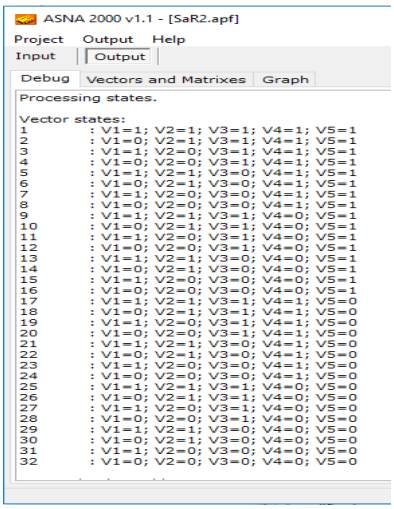

Fig. 1. The list of states with the value description of each component of vector states

$$Q_2(t) = P_8(t) + P_{16}(t) + \sum_{i=25}^{32} P_i(t) \qquad (6)$$

and as the sum of accident rate functions (7) $Q_1(t) + Q_2(t)$:

$$Q(t) = Q_1(t) + Q_2(t) = 1 - (1 - Q_1(t)) \cdot (1 - Q_2(t)) \qquad (7)$$

As it can be seen from Fig.2 the dependencies of both options to calculate the probability of accident rate occurrence coincided.

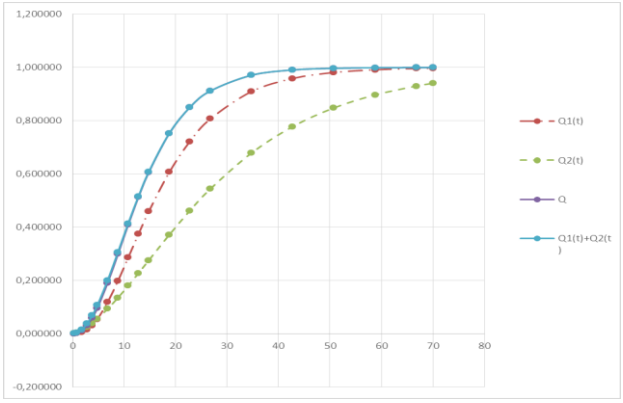

Fig. 2. The accident rate function $Q_1(t), Q_2(t)$ and the probability of accident rate occurrence $Q(t)$ – {curve $Q_1(t) + Q_2(t)$ covered curve $Q(t)$ }

The structural reliability schema was constructed using the graphical editor of ReliaSoft BlockSim program and is illustrated on Fig.3.

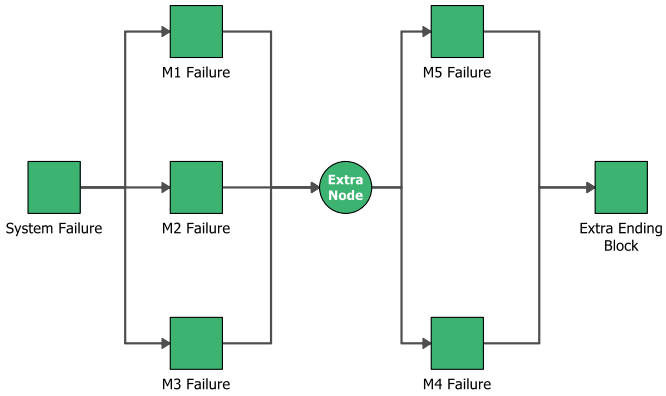

Fig. 3. The reliability block diagram obtained in ReliaSoft BlockSim

The next validation stage included the transformation of the structural reliability schema by means of ReliaSoft BlockSim to the fault-tolerant tree which is illustrated on Fig. 4. MCS were found for the fault-tolerant tree by means of ReliaSoft BlockSim, the probabilities of their occurrence were calculated at the same moments of time as well as accident rate functions and the comparison of results was made. As it can be seen from the Fig. 5 the value of the accident rate function $Q_1(t)$ and the value of probabilities of MCS1fta occurrence completely coincided. Similarly, the value of the accident rate function $Q_2(t)$ and the value of probabilities of MCS2 FTA occurrence coincided.

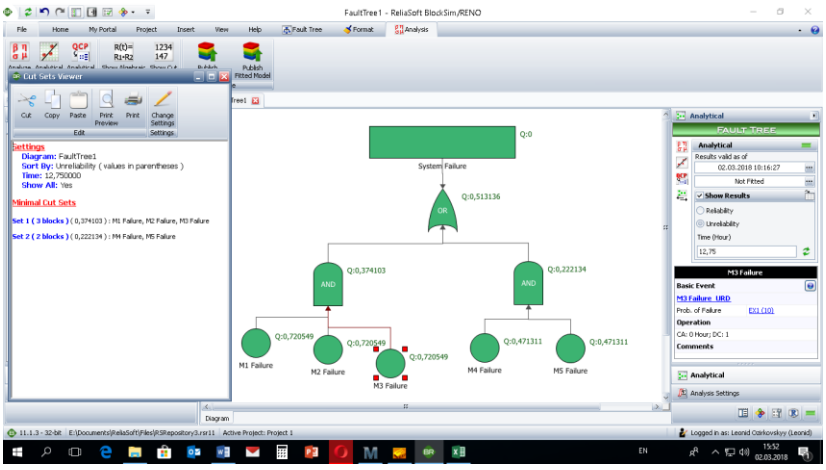

Fig. 4. The fault-tolerant tree

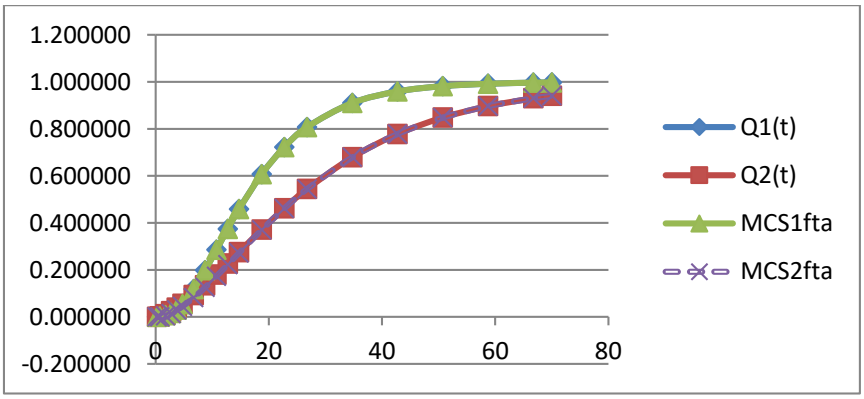

Fig. 5. The value of the accident rate function $Q_1(t)$ та $Q_2(t)$ and the value of the probabilities of MCS1fta and MCS2fta occurence

As a result, it can be concluded that the developed methods and methodology offer reliable results and it is possible to obtain the safety pointer using the graph of states and transitions.

6 Conclusions

1. The task of further research will be the development of behavior algorithms of universal reconnaissance complex for medium and unfavorable conditions and the study of their efficiency with considering the incorrect recognition of objects. The term ‘an accident rate function’ was introduced. It allowed to quantify the impact of reliability on safety and vice versa. It is seen that from a reliable model of the system in the form of a graph of states and transitions, it is possible to determine the accident rate function.

2. The confirmation of the accident rate function was provided by comparing two values – the value obtained from the accident rate function, at a determined interval of time with the values obtained from the occurrence probability of the minimum cut set. The minimum cut set was obtained from the fault tree for a similar time interval.

3. The proposed binary structural automata model with ASNA software allows to automate the design of fault-tolerant systems in the form of graphs of states and transitions, which is intended to determine the accident rate function.

4. The usage of ASNA software for the accident rate function and a new method to form accident rate functions from the subspace of the non-functioning states automates this process. Also, it allows a multivariate analysis without excessive time expenditures for the functional safety of the systems with responsible purpose.

References

1. US Department of Defense Standard Practice for System Safety: MIL-STD-882E, 101p. (2012)
2. US Department of Defense System Safety Program Requirements: MIL-STD-882C.-1993.
3. Bobalo Yu., Volochii B. Mathematical models and methods of reliability analysis of radioelectronic, electrical and software systems. O. Lozinsky, B. Mandzii, L. Ozirkovskii, D. Fedasyuk, S. Scherbovskikh, V. Yakovin. Lviv : Lviv Polytechnic Publishing House, 300 p. (2013). (in Ukrainian)
4. Kececioglu D. Reliability Engineering Handbook, Volume 2. Prentice Hall Inc.: New Jersey, 541 p. (1991).
5. Baldwin E. Carr Unmanned Aerial Vehicles: Examining the Safety, Security, Privacy and Regulatory Issues of Integration into U.S. Airspace. National Center for Policy Analysis, 44 p. (2013).
6. Myers A.: Complex System Reliability. Multichannel Systems with Imperfect Fault Coverage 2nd Edition. Springer-Verlag: London, 238 p. (2010).
7. Jan Krcál, Pavel Krcál Scalable Analysis of Fault Trees with Dynamic Features. 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks . (2015).
8. Yanbo Che, Yuancheng Zhao, Jianmei Xu, and Jinhuan Zhou / A Hierarchical Approach for Fast Calculating Minimal Cut Sets of a Microgrid. Mathematical Problems in Engineering, pp. 1 – 8. (2017).
9. Yuancheng Zhao, Yanbo Che, Tingjun Lin, Chuanyan Wang, Jiaxuan Liu, Jianmei Xu, Jinhuan Zhou Minimal Cut Sets-Based Reliability Evaluation of the More Electric Aircraft Power System // Mathematical Problems in Engineering, pp. 1 – 11. (2018).
10. Pang J., Liu Y., Mauw S.: Automatic Generation of Minimal Cut Sets //: 4th International Workshop on Engineering Safety and Security Systems 2015 (ESSS'15) EPTCS 184, pp. 33–47. (2015).
11. Guofeng Tang, Wei Gao Research of the Minimal Cut Sets Post Processing of the PSA Quantification Engine. Nuclear Safety, Security, Non-Proliferation and Cyber Security; Risk Management Shanghai, China, July 2–6, (2017).
12. Zuoyu Miao, Ru Niu : A new generation algorithm of fault tree minimal cut sets and its application in CBTC system. Tao Tang, Jieyu Liu. 2013 IEEE International Conference on Intelligent Rail Transportation Proceedings, pp. 11 - 23. (2013)
13. Woo Sik Jung A method to improve cutset probability calculation in probabilistic safety assessment of nuclear power plants // Reliability Engineering and System Safety, Volume 134, February, pp. 134-142. (2015).
14. Francesco Di Maio, Samuele Baronchelli, Enrico Zio Minimal Cut Sets Identification of Nuclear Systems by Evolutionary Algorithms. International Topical Meeting on Probabilistic Safety Assessment and Analysis, Sep 2013, Columbia, United States pp. 1-18 (2013).
15. Kara-Zaitri An improved minimal cut set algorithm // International Journal of Quality & Reliability Management, Vol. 13 Issue: 2, pp.114-132 (2013).
16. Dabrowski, C., Hunt, F. and Morrison, K., Improving the Efficiency of Markov Chain Analysis of Complex Distributed Systems. National Institute of Standards and Technology, Interagency Report 7744, 81 p. (2010).

17. Gandi Satyanarayana, Seetharamaiah P.: Component safety assessment using three-state Markov model. International Research Journal of Engineering and Technology (IRJET), Volume: 02 Issue: 09 | Dec-2015, pp. 299-306. (2015).
18. Vesely, W.E., Dugan, J., Fragola, J., Minarick III, J., Railsback, J.: Fault Tree Handbook with Aerospace Applications. National Aeronatics and Space Adminis-tration, August 2002. 218 p. (2002).
19. Eckard Bode, Thomas Peikenkamp : Model Based Importance Analysis for Minimal Cut Sets. Jan Rakow, Samuel Wischmeyer. International Symposium on Automated Technology for Verification and Analysis (ATVA 2008):Automated Technology for Verification and Analysis, 2008, P. 303-317. (2008).
20. US Department of Defense Standard Practice for System Safety: MIL-STD-882D.-2000
21. Block Sim. RBDs, fault trees and Markov diagrams. Режим доступу: https://www.reliasoft.com/products/reliability-analysis/blocksim
22. Volochiy B. Yu.: Method of Computation of Minimal Cut Sets of Fault-Tolerant Systems Based on Structural-Automatic Model. B. Yu. Volochiy, L. D. Ozirkovsky, A. V. Mashchak, O. P. Shkiliuk, I. V. Kulyk. Bulletin of National Technical University of Ukraine. Series Radiotechnique. Radioapparatus Building, № 52, pp. 38–45. (in Ukarinian).