PROBABILISTIC DEPENDABILITY ANALYSIS OF ADAPTIVE FUNCTIONS: A FAULT-TREE BASED APPROACH AND ITS APPLICATIONS IN TRANSPORTATION[1]

Géza SZABÓ* and Péter GÁSPÁR**

* Technical University of Budapest
H-1521 Budapest, Hungary
Tel: +36 1 463 1979; Fax: +36 1 463 3087
E-mail: szabo-g@kaut.kka.bme.hu
** Computer and Automation Research Institute
Kende u. 13-17
H-1111 Budapest, Hungary
Tel: (+36 1) 466 7483; Fax: (+36 1) 466 75 03
E-mail: gaspar@sztaki.hu

Received: October 21, 1998

Abstract

In this paper, a new method is proposed as an extension of the classic fault-tree analysis for the treatment of functionally re-configurable logic. Such logic may change its operation if a failure has occurred and been detected. Since this logic can distinguish between the detected failure and the undetected failure, these must be considered in the fault-tree structure. Closed formulas are applied to ensure efficient algorithm for the analysis. The paper summarizes that way which enables the treatment of re-configuration, and analyzes the limitations of the described methods.

Keywords: fault-tree analysis, reliability analysis, re-configurable component, fault-tolerant system.

1. Introduction

In fault-tolerant, safety-oriented industrial systems, increasing availability is one of the most important objectives. General techniques for increasing availability of systems include the use of more reliable components; the application of more redundancy, etc. The dependability of the system can be examined by both deterministic and probabilistic ways. In the deterministic analysis the single failure criterion is examined, or other attributes of dependability, such as maintainability, testability, etc. can be analyzed. In the probabilistic analysis, availability is measured by the system operation

[1]This project has been supported by the Hungarian National Science Foundation through grant no. T-016418.

without malfunction, or unavailability is calculated as a probability or a frequency of a non-desired system function. The system must reach certain predefined limits, which are set by authorities or based on experience from previous developments.

Nowadays, the application of fault-adaptive, functionally re-configurable components is a frequently used method for increasing availability. This type of component may change its function if the failure has occurred and been detected in order to neglect the failure and prevent its spreading. The condition for this ability is the information about the failure occurred, which can be provided, e.g. by a status signal from a failure detection mechanism.

The fault-tree analysis (FTA) is widely used in industrial systems, e.g. power generation systems, vehicle systems, because its methodology is well elaborated and a large number of efficient analysis procedures exist, (BOKOR et al. 1997; BROW, 1990). The objective of the FTA is to calculate the availability or unavailability of the system, but other useful information can also be gained about the structure of the system, (APOSTOLAKIS et al. 1978; CHUNNING and DINGHUA, 1990; SCHNEEWEIS, 1985; STECHER, 1986). Besides the probability of the analyzed top event, the traditional FTA can also provide other useful information such as the importance of components, the sensitivity of components and a set of basic events, called MCS, which can be selected according to the following. The top event will occur if all basic events of the MCS have occurred, but if at least one basic event does not occur the top event will not occur (HWANG et al. 1981; LEE et al. 1985). Thus, the minimal cut sets show how many individual component failures are needed for the system failure, so the weak points of the system can be determined, thus the single failure criterion can also be analyzed based on fault- trees. It can be investigated how the availability of the system increases (or decreases) if failure rates of different components increase (or decrease) which is the basis of the importance analysis. The components, to which the availability of the system is extremely sensitive and also the kind of components to which it is not sensitive can be selected.

One of the most important difficulties in FTA is the treatment of dynamic components, e.g. hot and cold spares, priority logic, etc. In the literature, several methods are developed to take these components into account in FTA, e.g. DOYLE et al., 1995; GULATI and DUGAN, 1997; KAI, 1990.

In this paper, the treatment method of re-configurable components is proposed. Moreover, the three-state model is introduced instead of the two-state model in the traditional FTA. When a re-configurable component is modeled, the two states (event has been occurred or not) are not adequate, since the component has three possible states: the failure has occurred and been detected, the failure has occurred and not been detected and the failure

[1]This project has been supported by the Hungarian National Science Foundation through grant no. T-016418.

has not occurred. Traditional tools can also handle the problem by dividing the detected and undetected failures and modeling the adaptivity with the basic traditional gates, but this method causes certain difficulties (GÁSPÁR and SZABÓ, 1998).

The structure of the paper is as follows. In Chapter 2, the definition of the active and the adaptive logic is introduced, and the handling of this logic is illustrated by the so-called macro models. In Chapter 3, closed formulas are suggested to ensure unified treatment of the gates with different number of inputs. In Chapter 4, a demonstration example is illustrated for the handling of the adaptive logic and several application subjects are outlined from transportation engineering.

2. Definition and Treatment of Re-Configurable Logic

2.1. Definition of Active and Adaptive Logic

An average system can consist of the following operation modes or functions.

- Normal functions: These functions cannot distinguish the detected failure from the undetected failure.
- Active functions: These functions send forward information about the success of the function execution or the fault. However, these functions cannot change their actually executed functions according to this information.
- Adaptive functions: These functions can change their actual function dynamically according to the faulty input or failure status and can provide information about the success of the execution. E.g. in the case of an adaptive AND function with 3 redundant inputs, the first detected failure causes a re-configuration to another AND function with 2 inputs and the failed input is blocked. In the case of an adaptive 2-out-of-3 (2v3) logic, the first detected failure causes a re-configuration of the function, which can be modeled with a 2v2 logic. Moreover, the second detected failure causes further re-configuration to an AND logic with one valid input and two blocked inputs.

2.2. Macro Models: a Traditional Solution for Modeling the Re-Configuration

The traditional fault-tree analysis is capable of modeling the active or adaptive system functions. Since for adequate modeling the detected and the undetected failures must be distinguished at the basic event level, and since the fault-tree models of active and adaptive components also require the distinction of these failure modes, every event in a fault-tree model must be described by two, separated probabilities.

A macro model for active and adaptive functions cannot be set up by using only the traditional fault-tree gates (AND, OR, K-out-of-N). This modeling requires the supplementary gates of the traditional fault-tree (NOT, XOR). *Fig. 1* shows the macro model of a normal 2v3 function, *Fig. 2* shows the active and *Fig. 3* the adaptive ones.

The macro model solution has several disadvantages:

- As the number of inputs increase, the macro model will become extremely complex.
- If macro models are used at different levels of the fault-tree, the analysis will require a long period of time.
- The models cannot handle the relationships between detected and undetected failures.

2.3. Closed Formulas for the Re-Configurable Logic in FTA

The calculation of probability of the different states can be formalized in such a way that only the probabilities of the detected failure and the undetected failure are used. In the following, closed formulas are presented in the case of different types of gates, namely the OR gate, the AND gate, and the KvN gate.

An OR gate has an output failure if at least one of the inputs has a failure. If the logic cannot distinguish between detected and undetected failures on the input side, the probability of the output failure can be calculated by using a simple OR formula of the failures. This value is considered as an undetected failure at the next level. The information that an input failure has occurred may be used in the re-configuration step of this logic in order to realize a new function using the other inputs, which are assumed to be error-free. From the point of view of failures, the active logic can distinguish between detected failures and undetected failures, however, because this type cannot change the function the sum of the two probabilities is the same as the probability of the normal OR gate. In the case of adaptive logic, the first detected failure causes a re-configuration of this gate to another OR gate with the other inputs, and the failed input is blocked. Let P_{Di} be the probability of the detected failure of the i^{th} input, and P_{Ni} be the probability of the undetected failure. The formulas of probabilities of detected and undetected failures are shown in *Table 1*.

In the case of an AND gate, the advantages shown in the previous paragraph do not arise, because a failure only occurs on the output if detected failures occur on all of the inputs. In the case of an active AND gate, detected failure is indicated on the output if all of the inputs have detected failure. It is unnecessary to define the adaptive type of this gate because if the input signal with detected failure is omitted from the calculation, the type of function is not affected. The formulas of probabilities of detected and undetected failures are shown in *Table 2*.

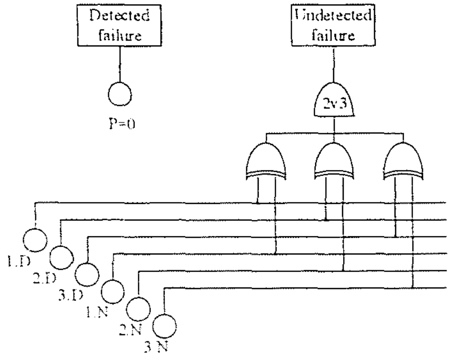

Fig. 1. Illustration of normal 2v3 logic using the macro model

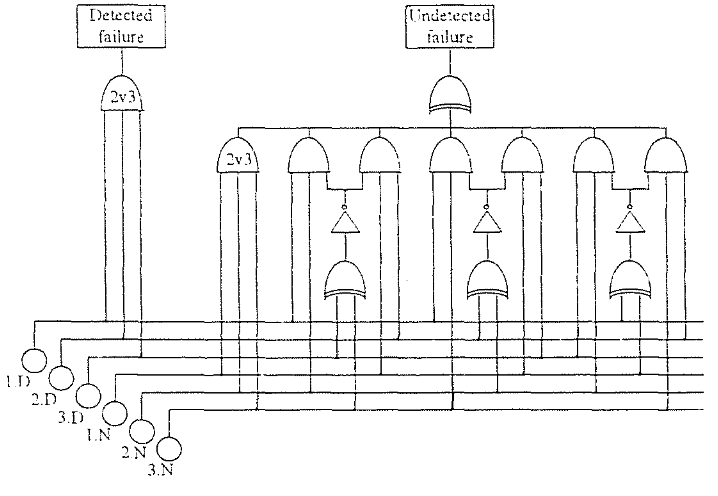

Fig. 2. Illustration of active 2v3 logic using the macro model

The traditional KvN gate cannot distinguish between detected and undetected failures, the output failure probability is treated as undetected failure probability, however, active and adaptive logic can distinguish between detected and undetected failures. E.g. in the case of the adaptive 2v3 gate, the first detected failure causes a re-configuration that can be modeled with a 2v2 gate. The effect of the second detected failure is a further re-configuration to a gate that passes the third input to the output. The formulas of probabilities of detected and undetected failures are shown in *Table 3*. As an illustration, the framework of the proofs is shown in the

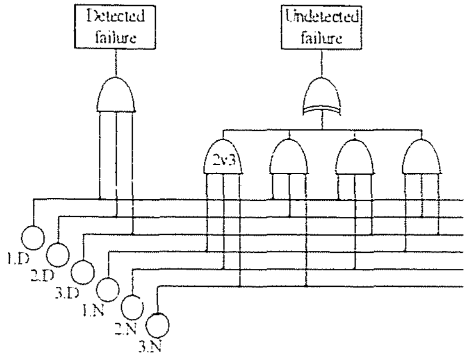

Fig. 3. Illustration of adaptive 2v3 logic using the macro model

Appendix for the creation of the closed formulas in the case of three inputs.

Table 1. The probability of detected and the undetected failures of the OR gate

Failure	Equation
Normal	$P_{\text{OR}}^{\text{Normal}} = \sum_{m=1}^{N}\left[(-1)^{m+1}\sum_{\substack{P_{A\ell_j}\in\{P_{D\ell_j},P_{N\ell_j}\}\\ \ell_j\in\{1\ldots N\}:\ell_1<\ldots<\ell_m}}\prod_{j=1}^{m}\left(P_{A\ell_j}\right)\right]$
Active detected	$P_{\text{Det,OR}}^{\text{Active}} = \sum_{m=1}^{N}\left[(-1)^{m+1}\sum_{\ell_j\in\{1\ldots N\}:\ell_1<\ldots<\ell_m}\prod_{j=1}^{m}\left(P_{D\ell_j}\right)\right]$
Active undetected	$P_{\text{Undet,OR}}^{\text{Active}} = \sum_{m=1}^{N}\left[(-1)^{m+1}\sum_{\substack{P_{A\ell_j}\in\{P_{N\ell_j},P_{D\ell_j}\}\\ \ell_j\in\{1\ldots N\}:\ell_1<\ldots<\ell_m\\ \exists\ell_s\in\{\ell_1\ldots\ell_m\}:P_{A\ell_s}=P_{N\ell_s}}}\prod_{j=1}^{m}\left(P_{A\ell_j}\right)\right]$
Adaptive detected	$P_{\text{Det,OR}}^{\text{Adaptive}} = \prod_{j=1}^{N}\left(P_{D_j}\right)$
Adaptive undetected	$P_{\text{UnDet,OR}}^{\text{Adaptive}} = \sum_{m=1}^{N}\left[(-1)^{m+1}\sum_{\ell_j\in\{1\ldots N\}:\ell_1<\ldots<\ell_m}\prod_{j=1}^{m}\left(P_{N\ell_j}\right)\right]$

Table 2. The probability of detected and the undetected failures of the AND gate

Failure	Equation
Normal	$P_{\text{AND}}^{\text{Normal}} = \sum_{P_{Aj} \in \{P_{Nj}, P_{Dj}\}} \left[\prod_{j=1}^{N} (P_{Aj}) \right]$
Active detected	$P_{\text{Det,AND}}^{\text{Active}} = \prod_{j=1}^{N} (P_{Dj}) = P_{\text{Det,AND}}^{\text{Adaptive}}$
Active undetected	$P_{\text{Undet,AND}}^{\text{Active}} = \sum_{\substack{P_{Aj} \in \{P_{Nj}, P_{Dj}\} \\ \exists j : P_{Aj} = P_{Nj}}} \left[\prod_{j=1}^{N} (P_{Aj}) \right] = P_{\text{Un det,AND}}^{\text{Adaptive}}$

Table 3. The Probability of detected and the undetected failures of the KvN gate

Failure	Equation
Normal	$P_{\text{KvN}}^{\text{Norm}} = \sum_{m=K}^{N} \left[(-1)^{m+K} \binom{m-1}{K-1} \sum_{\substack{P_{A\ell_j} \in \{P_{D\ell_j}, P_{N\ell_j}\} \\ \ell_j \in \{1 \ldots N\} : \ell_1 < \ldots < \ell_m}} \prod_{j=1}^{m} (P_{A\ell_j}) \right]$
Active detected	$P_{\text{Det,KvN}}^{\text{Active}} = \sum_{m=K}^{N} \left[(-1)^{m+K} \binom{m-1}{K-1} \sum_{\ell_j \in \{1 \ldots N\} : \ell_1 < \ldots < \ell_m} \prod_{j=1}^{m} (P_{D\ell_j}) \right]$
Active undetected	$P_{\text{Undet,KvN}}^{\text{Active}} = \sum_{m=K}^{N} \left[(-1)^{m+K} \binom{m-1}{K-1} \sum_{\substack{P_{A\ell_j} \in \{P_{N\ell_j}, P_{D\ell_j}\} \\ \ell_j \in \{1 \ldots N\} : \ell_1 < \ldots < \ell_m \\ \exists \ell_s \in \{\ell_1 \ldots \ell_m\} : P_{A\ell_s} = P_{N\ell_s}}} \prod_{j=1}^{m} (P_{A\ell_j}) \right]$
Adaptive detected	$P_{\text{Det,KvN}}^{\text{Adaptive}} = \prod_{j=1}^{N} (P_{\text{D}j})$
Adaptive undetected	$P_{\text{Undet,KvN}}^{\text{Adaptive}} = \sum_{m=K}^{N} \left[(-1)^{m+K} \binom{m-1}{K-1} \sum_{\ell_j \in \{1 \ldots N\} : \ell_1 < \ldots < \ell_m} \prod_{j=1}^{m} (P_{N\ell_j}) \right] + \sum_{\substack{P_{A\ell_j} \in \{P_{N\ell_j}, P_{D\ell_j}\} \\ \ell_j \in \{1 \ldots N\} : \ell_1 < \ldots < \ell_m \\ \exists \ell_s \in \{\ell_1 \ldots \ell_k\} : P_{A\ell_s} = P_{N\ell_s} \\ \forall \ell_z \in \{\ell_1 \ldots \ell_k\} : \ell_z \neq \ell_s : P_{A\ell_z} = P_{D\ell_z}}} \prod_{j=1}^{N} (P_{A\ell_j})$

3. Application of the Theory

3.1. Demonstration Example

In the example, a part of a coach door control system will be analyzed for reliability. The system contains three sensors for each door to sense the open or closed state of the door (triple redundancy), and the sensors are connected to a processor-based central controller via an RS-485 bus, see *Fig. 4*. Among others the tasks of the central controller are to observe the door state (open or closed); to prevent the coach from departing with open doors; and if one of the doors opens while the vehicle is moving, to decrease the speed of the coach to zero by actuating the brake system.

The central controller processes the signals from sensors by majority voting, in this case a 2v3 logic. The processing mode is normal, if any failure of two of the three sensors causes a malfunction of the central controller. In our example, a failure of the vehicle's central bus causes the loss of the three sensor signals, consequently, the activation of brakes. Adaptive (or re-configurable) processing mode is applied if the detected failures in the input information (the signals coming from the sensors) of the central controller have a different effect on the system than the undetected failures. In our case, the signal of a sensor with a detected failure can be ignored, consequently, the system can operate with two detected failures as well as with good signals. If all of the input signals of the central controller fail, we have the ability to choose the solution with the knowledge of malfunction in the system.

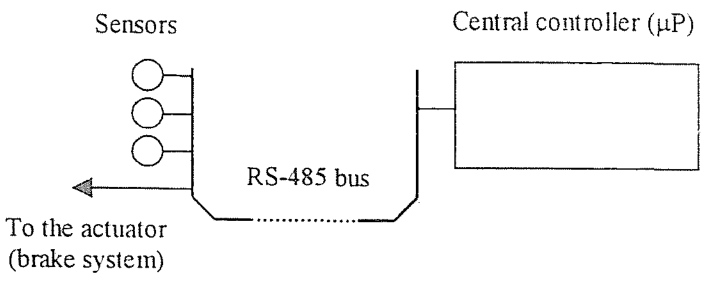

Fig. 4. Coach door control system (simplified) for demonstration purposes

Besides the two operating modes mentioned above there is a third mode: it does not provide all of the advantages of the adaptive mode, but provides the ability to detect a malfunction. The operation mode is active if the logic does not take the statuses of the signals (failed or not) into account when calculating the output signal (the result), but based on the statuses, it calculates an output status information, which signals if the result cannot be valid due to the detected failures (e.g. two detected failures in a 2v3 logic). This information can be applied in large systems for further processing.

Below we will demonstrate the solutions described in the paper for adaptive components. *Fig. 5* shows a fault-tree built for the system ignoring the adaptive function. This fault tree is very simple and a traditional 2v3 gate represents the voting function. Of course the results from an analysis of this fault tree will be different from the real values because of ignoring, but in some cases the difference is acceptable. This method can be applied when the probability of detected failures is much less than the probability of undetected failures. As shown in *Fig. 5*, the detected and undetected failures are not distinguished in this case and treated together as a single failure of an element.

In *Fig. 6* an application of macro models is presented. The fault tree uses a macro model not only for adaptive 2v3 logic, but for the logical connection of bus and sensor failures. This logical connection can be modeled as an active OR function: if one of the two components considered (one sensor and the bus) has a detected failure, the central controller can change its function. Because this is a logical connection of failures and not a real function, it cannot be adaptive.

Of course, the application of macro models can be simplified on the fault-tree as well as we can simplify the fault trees for closed formulas, using special gates for the drawing. This is shown in *Fig. 7*. In this case, the lines in the fault tree represent the detected and the undetected failures and the events have three states as described in the previous part of the paper.

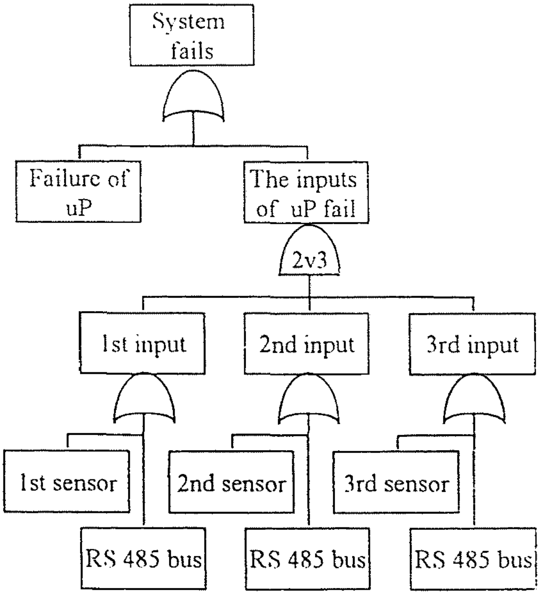

Fig. 5. Traditional fault tree of the system

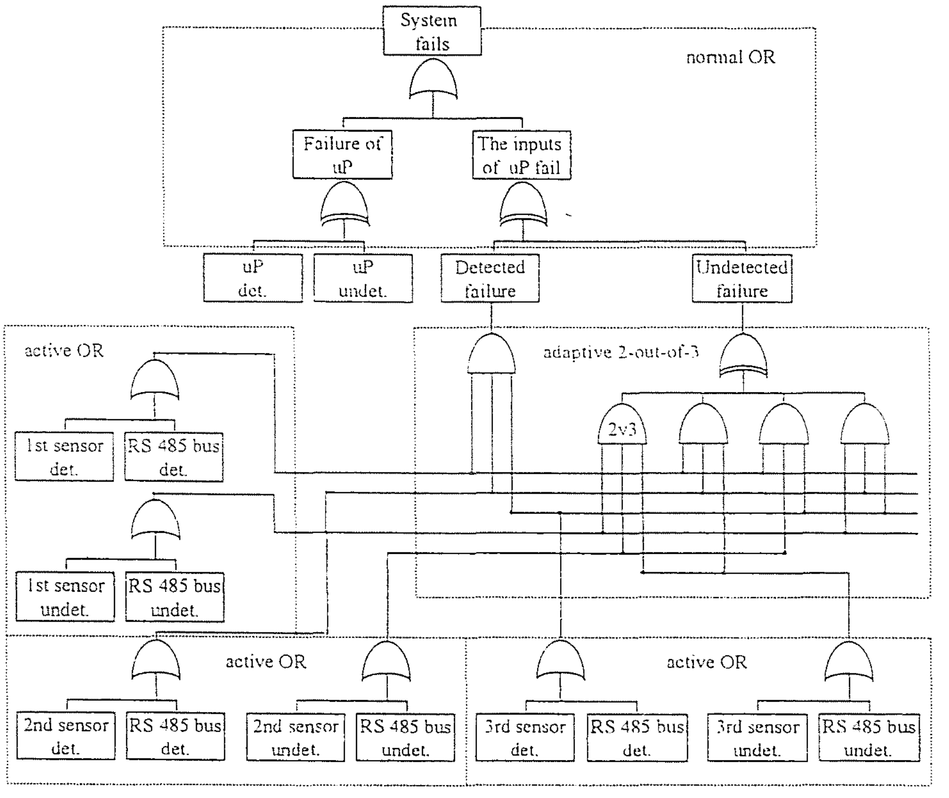

Fig. 6. Application of macro models

3.2. Some Applications in Traffic Automation

The main areas of application for fault-tree analysis in transportation are the aviation, space and vehicle technology, where electrical and mechanical components must be analyzed together, (DUGAN et al., 1990; CHARLESWORTH and RAO, 1985). Aviation and space technology require fault-tolerant systems because no safety state of the machine or its control subsystems can be found. The speciality of the space vehicles is that during the mission time (or the whole life-cycle of the components) it is not possible to maintain the components. Testing and failure detection are also important in these cases, because switching to the spare unit requires the knowledge of a fault and (mainly at the end of the life-cycle, when spares are no longer available) fault-adaptive techniques can ensure a longer life-cycle.

The availability analysis becomes more important in some non-safety oriented areas in transportation because of the increasing demands of customers. Truck and van central braking systems are good example of this, (LIMING et al., 1996; HUDOKLIN and ROZMAN, 1985).

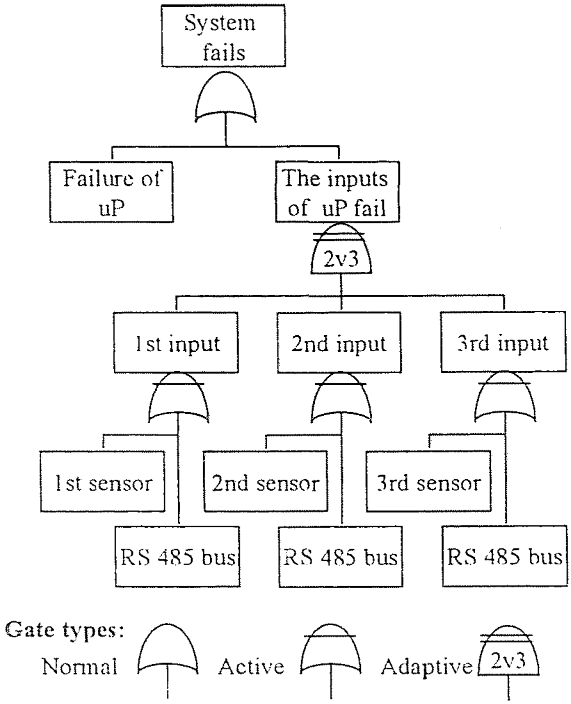

Fig. 7. Fault-tree for closed formulas

Conclusion

In this paper we have introduced closed formulas ensuring the treatment of functionally re-configurable logic in fault-tree analysis and we have also showed the application in a demonstration example. The formulas are valid for any number of inputs, thus the detected and the undetected failure probability can be calculated, which is a more efficient way than developing the related macro models and incorporating them into the fault-tree. It seems necessary to use the extended fault-tree analysis in the reliability analysis of transportation systems.

References

[1] APOSTOLAKIS, G. – GARRIBBA, S. – VOLTA, G. (Eds.): Synthesis and Analysis Methods for Safety and Reliability Studies, Plenum, 1978.

[2] BOKOR, J. – SZABÓ, G. – GÁSPÁR, P. – HETTHÉSSY, J.: Reliability Analysis of Protection Systems in NPP Applying Fault-Tree Analysis Method. *Proc. of the Computerized Reactor Protection and Safety Related Systems in Nuclear Power Plants*, Budapest, Hungary, 1997. pp. 91–104.

[3] BROW, K. S.: Evaluating Fault Trees (and & or Gates Only) with Repeated Events, *IEEE Transactions on Reliability*, Vol. 39, No. 2., 1990, pp. 226–235

[4] CHARLESWORTH, W. W. – RAO, S. S.: Reliability Analysis of Continuous Mechanical Systems Using Multistate Fault Trees, *Reliability Engineering and System Safety*, Vol. 37, No. 3., 1985, pp. 195–206.

[5] CHUNNING, Y. – DINGHUA, S.: Classification of Fault Trees and Algorithms of Fault Tree Analysis, *Microelectronics and Reliability*, Vol. 30, No. 5, 1990, pp. 891–895.

[6] DOYLE, S. A. – DUGAN, J. B. – BOYD, M.: Combinatorical Models and Coverage: a Binary Decision Diagram Approach, *Proc. of the Annual Reliability and Maintainability Symposium*, 1995, pp. 82–89.

[7] DUGAN, J. B. – BAVUSO, S. J. – BOYD, M. A.: Fault Trees and Sequence Dependencies, *Proc. of the Annual Reliability and Maintainability Symp.*, 1990, pp. 286–293.

[8] GÁSPÁR, P. – SZABÓ, G.: Analysis of Adaptive Multi-State Logic in Fault-Tolerant Systems, International Conference on PSAM 4, New York, 1998.

[9] GÁSPÁR, P. – BOKOR, J. – NAGY, I.: Control of Large Systems in a Nuclear Power Plant, *Proc. of the MOVIC*, Yokohama, Vol. 1, 1994, pp. 377–382.

[10] GULATI, R. – DUGAN, J. B.: A Modular Approach for Analyzing Static and Dynamic Fault Trees, *Proc. of the Annual Reliability and Maintainability Symp.*, 1997, pp. 57–63.

[11] HUDOKLIN, A. – ROZMAN, V.: Safety Analysis of the Railway Traffic System, *Reliability Engineering and System Safety*, Vol. 37, No. 3., 1985, pp. 7–13.

[12] HWANG, C. L. – TILLMAN, F. A. – LEE, M. H.: System-Reliability Evaluation Techniques for Complex/Large Systems – A review, *IEEE Trans. on Reliability*, Vol. R-30, No. 5, 1981, pp. 416–423.

[13] KAI, YU.: Multistate Fault-Tree Analysis, *Reliability Engineering and System Safety*, Vol. 28, 1990, pp. 1–7.

[14] LEE, W. S. – GROSH, D. L. – TILLMAN, F. A. – LIE, C. H.: Fault Tree Analysis, Methods and Applications – A Review, *IEEE Transactions on Reliability*, Vol. R-34, No. 3., 1985, pp. 194–203.

[15] LIMING, J. K. – LOH, W. T. – DONELSON, A. C.: Scenario-Based Probabilistic Safety Assessment Techniques for Automobiles and Trucks, *International Conference on PSAM 3*, Athens, 1996, pp. 1422–1427.

[16] SCHNEEWEIS, W. G.: Fault-Tree Analysis Using a Binary Decision Tree, *IEEE Transactions on Reliability*, Vol. R-34, No. 5., 1985, pp. 453–457.

[17] STECHER, K.: Evaluation of Large Fault-Trees with Repeated Events Using an Efficient Bottom-Up Algorithm, *IEEE Transactions on Reliability*, Vol. R-35, No. 1., 1986, pp. 51–58.

Appendix

$$
\begin{aligned}
P_{2v3}^{\text{Normal}} &= P_{N_1}P_{N_2} + P_{N_1}P_{N_3} + P_{N_1}P_{D_2} + P_{N_1}P_{D_3} + P_{D_1}P_{N_2} + \\
&\quad + P_{D_1}P_{N_3} + P_{N_2}P_{N_3} + P_{N_2}P_{D_3} + P_{D_2}P_{N_3} + P_{D_1}P_{D_2} + \\
&\quad + P_{D_1}P_{D_3} + P_{D_2}P_{D_3} - 2P_{N_1}P_{N_2}P_{N_3} - 2P_{D_1}P_{D_2}P_{N_3} - \\
&\quad - 2P_{D_1}P_{N_2}P_{N_3} - 2P_{D_1}P_{N_2}P_{D_3} - 2P_{N_1}P_{D_2}P_{N_3} - \\
&\quad - 2P_{N_1}P_{N_2}P_{D_3} - 2P_{N_1}P_{D_2}P_{D_3} - 2P_{D_1}P_{D_2}P_{D_3} \\
&= \sum_{\substack{P_{A\ell_1}, P_{A\ell_2} \in \{P_{D\ell}, P_{N\ell}\} \\ \ell_1, \ell_2 \in \{1\ldots3\}, \ell_1 < \ell_2}} P_{A\ell_1}P_{A\ell_2} - \\
&\quad - \sum_{\substack{P_{A\ell_1}, P_{A\ell_2}, P_{A\ell_3} \in \{P_{D\ell}, P_{N\ell}\} \\ \ell_1, \ell_2, \ell_3 \in \{1\ldots3\}, \ell_1 < \ell_2 < \ell_3}} 2P_{A\ell_1}P_{A\ell_2}P_{A\ell_3} \\
&= \sum_{m=2}^{3} \left[(m-1)(-1)^m \sum_{\substack{P_{A\ell_j} \in \{P_{D\ell_j}, P_{N\ell_j}\} \\ \ell_j \in \{1\ldots3\}, \ell_1 < \ldots < \ell_m}} \prod_{j=1}^{m} \left(P_{A\ell_j} \right) \right] \\
P_{\text{Det},2\text{v}3}^{\text{Active}} &= P_{D_1}P_{D_2} + P_{D_1}P_{D_3} + P_{D_2}P_{D_3} - 2P_{D_1}P_{D_2}P_{D_3} \\
&= \sum_{\ell_1, \ell_2 \in \{1\ldots3\}, \ell_1 < \ell_2} P_{D\ell_1}P_{D\ell_2} - 2P_{D_1}P_{D_2}P_{D_3} \\
&= \sum_{m=2}^{3} \left[(m-1)(-1)^m \sum_{\ell_j \in \{1\ldots3\}, \ell_1 < \ldots < \ell_m} \prod_{j=1}^{m} \left(P_{D\ell_j} \right) \right] \\
P_{\text{UnDet},2\text{v}3}^{\text{Active}} &= P_{N_1}P_{D_2} + P_{N_1}P_{D_3} + P_{D_1}P_{N_2} + P_{D_1}P_{N_3} + P_{N_2}P_{D_3} + \\
&\quad + P_{D_2}P_{N_3} + P_{N_1}P_{N_2} + P_{N_1}P_{N_3} + P_{N_2}P_{N_3} - \\
&\quad - 2P_{N_1}P_{N_2}P_{N_3} - 2P_{D_1}P_{D_2}P_{N_3} - 2P_{D_1}P_{N_2}P_{N_3} - \\
&\quad - 2P_{D_1}P_{N_2}P_{D_3} - 2P_{N_1}P_{D_2}P_{N_3} - 2P_{N_1}P_{N_2}P_{D_3} - \\
&\quad - 2P_{N_1}P_{D_2}P_{D_3} \\
&= \sum_{\substack{P_{A\ell_1}, P_{A\ell_2} \in \{P_{D\ell}, P_{N\ell}\} \\ \ell_1, \ell_2 \in \{1\ldots3\}, \ell_1 < \ell_2}} P_{A\ell_1}P_{A\ell_2} - \\
&\quad - \sum_{\substack{P_{A\ell_1}, P_{A\ell_2}, P_{A\ell_3} \in \{P_{D\ell}, P_{N\ell}\} \\ \ell_1, \ell_2 \ell_3 \in \{1\ldots3\}, \ell_1 < \ell_2 < \ell_3 \\ \exists \ell_s \in \{\ell_1, \ell_2, \ell_3\} : P_{A\ell_s} = P_{N\ell_s}}} 2P_{A\ell_1}P_{A\ell_2}P_{A\ell_3}
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{m=2}^{3}\left[(m-1)(-1)^m \sum_{\substack{P_{A\ell_j}\in\{P_{N\ell_j},P_{D\ell_j}\}\\ \ell_j\in\{1\ldots3\},\ell_1<\ldots<\ell_m\\ \exists\ell_s\in\{\ell_1\ldots\ell_m\}:P_{A\ell_s}=P_{N\ell_s}}} \prod_{j=1}^{k}\left(P_{A\ell_j}\right)\right]\\
P^{\text{Adaptive}}_{\text{Det},2\text{v}3} &= P_{D_1}P_{D_2}P_{D_3} = \prod_{j=1}^{3}\left(P_{D_j}\right)\\
P^{\text{Adaptive}}_{\text{UnDet},2\text{v}3} &= P_{N_1}P_{N_2}+P_{N_1}P_{N_3}+P_{N_2}P_{N_3}+P_{D_1}P_{D_2}P_{N_3}+\\
&\quad +P_{D_1}P_{N_2}P_{D_3}+P_{N_1}P_{D_2}P_{D_3}-2P_{N_1}P_{N_2}P_{N_3}\\
&= \sum_{\ell_1,\ell_2\in\{1\ldots3\},\ell_1<\ell_2} P_{N\ell_1}P_{N\ell_2}+\\
&\quad + \sum_{\substack{P_{A\ell_1},P_{A\ell_2},P_{A\ell_3}\in\{P_{D\ell},P_{N\ell}\}\\ \ell_1,\ell_2,\ell_3\in\{1\ldots3\}:\ell_1<\ell_2<\ell_3\\ \exists\ell_s\in\{\ell_1,\ell_2,\ell_3\}:P_{A\ell_s}=P_{N\ell_s}\\ \forall\ell_z\in\{\ell_1,\ell_2,\ell_3\}:\ell_z\neq\ell_s:P_{A\ell_z}=P_{D\ell_z}}} P_{A\ell_1}P_{A\ell_2}P_{A\ell_3}-\\
&\quad -2P_{N_1}P_{N_2}P_{N_3}\\
&= \sum_{m=2}^{3}\left[(m-1)(-1)^m \sum_{\ell_j\in\{1\ldots3\}:\ell_1<\ldots<\ell_m} \prod_{j=1}^{m}\left(P_{N\ell_j}\right)\right]+\\
&\quad + \sum_{\substack{P_{A\ell_j}\in\{P_{N\ell_j},P_{D\ell_j}\}\\ \ell_j\in\{1\ldots3\}:\ell_1<\ldots<\ell_3\\ \exists\ell_s\in\{\ell_1,\ldots,\ell_3\}:P_{A\ell_s}=P_{N\ell_s}\\ \forall\ell_z\in\{\ell_1\ldots\ell_3\}:\ell_z\neq\ell_s:P_{A\ell_z}=P_{D\ell_z}}} \prod_{j=1}^{3}\left(P_{A\ell_j}\right)
\end{aligned}
$$