Overview of risk assessment in digitalized nuclear power plants

KANG Hyun Gook[1], and SHIN Sungmin[2]

1. Department of Nuclear and Quantum Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea (hyungook@kaist.ac.kr)

2. Department of Nuclear and Quantum Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea (becomejade@kaist.ac.kr)

Abstract: Many of non-safety analog features in nuclear power plants (NPP) are being replaced with digitalized systems to gain advantages in accuracy, computational capability, and data handling. It is difficult however to ensure the safety of digital features because there is, as of yet, no comprehensive reliability quantification method for them. In this overview, preceding studies related to three critical factors in the reliability quantification process, namely detection coverage of fault–tolerant techniques, software reliability, and network communication failure, are introduced and their valuable insights and challenges are described.

Keywords: digital safety systems; detection coverage; fault-tolerant techniques; software reliability; network failure; digitalized NPP

1 Introduction

Over the past few decades, various digital systems have been supplanting the analog systems in nuclear power plants (NPP) to utilize the advanced digital features. A report published in 1997 by the U.S. National Research Council states that appropriate methods for assessing safety and reliability are key to establishing the acceptability of digital instrumentation and control (I&C) systems in safety-critical plants such as NPPs [1]. Since the release of this report, the development of a methodology for the probabilistic safety assessment (PSA) of digital I&C systems has been a critical issue. However, there is still no widely accepted method [2]. Kang and Sung found that the detection coverage of fault-tolerant techniques, software reliability, and network communication failure are the three most critical factors in the safety assessment of digital systems [3]. In this overview, recent noteworthy approaches and challenging points for each of these factors are briefly introduced.

This paper is organized as follows. In Section 2, brief concept for detection coverage of fault-tolerance techniques and a simulation based approach for quantification of the coverage are introduced. Then, limitations and challenges of two representative test based approaches for software reliability quantification are described in Section 3. In Section 4, a study taking comprehensive approach for quantification of network communication risk is introduced.

2 Detection coverage of fault tolerance techniques

Fault tolerance is the capability of a system to work properly in spite of the existence of faults. In comparison with traditional analog systems, digitalized systems have more diverse fault-tolerant techniques to improve system safety. As there is no proper basis so far to obtain digital system reliability regarding fault tolerance, proper evaluation methods need to be developed. Fault detection coverage, which is the ability to detect errors, is considered as one of the most crucial factors in the assessment process, as a system can fail when faults go undetected by the adapted fault-tolerant techniques. The number indicating the coverage is directly connected to system safety, which is the overall goal of current reliability quantification research.

2.1 Characteristics of fault tolerance techniques in digital I&C systems

All possible faults in a system cannot be detected by any one specific fault-tolerant technique, as each technique merely covers a certain range of faults. Therefore, multiple fault-tolerant techniques are applied at several levels of system hierarchy to achieve better reliability. By doing so, even if a fault is not detected by one technique in a lower level, it can

Received date: March 9, 2016
(Revised date: March 21, 2016)

be detected by another one at a higher level. Figure 1 shows this conceptual structure of multiple fault-tolerant techniques. Examples of conventional techniques in each level are memory check sum and watchdog timer on the component level; loop back check for input and output modules on the board level; and automatic periodic testing and state comparison algorithm of redundant modules on the system level.

Respective fault-tolerant techniques not only have different ranges of inspection but also different inspection periods, from almost continuous monitoring to monthly inspection. Therefore, the different inspection range and period of each technique should be properly considered to exclude duplicated effects for the appropriate evaluation of fault detection coverage.

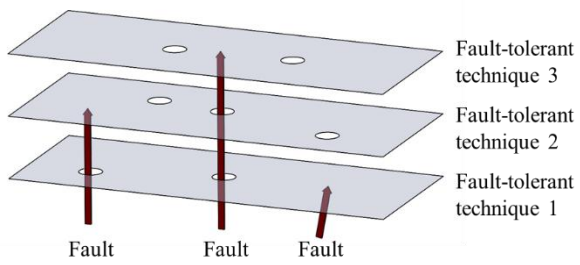

Fig. 1 Faults and fault-tolerant techniques.

2.2 Quantification of fault detection coverage

When there are multiple fault-tolerant techniques on several levels of a system, a fault which is not detected by one technique could be detected by another, or by a number of different techniques concurrently (duplicated effect). This leads to the overall fault detection coverage not being a simple summation of each technique's coverage but a union set of all techniques. In order to exclude duplicated effects, the relations between faults and fault-tolerant techniques need to be precisely identified. Then the definition of fault detection coverage can be mathematically expressed as a conditional probability that gives the existence of a fault [2, 4].

The fault detection coverage of a union of fault-tolerant techniques can be identified through the fault injection experiment. For this experiment, S.J. Lee *et al.* [5] considered the following three steps: identify all possible faults in a target system, determine a proper simulation method based on the given experimental environment, and perform the fault injection experiment. For fault identification, the failure mode and effects analysis (FMEA) method is utilized. Through this method, two failure cases are categorized: safe failure, where there is no effect and the system works normally, and dangerous failure, where effects cause an abnormal status of the system. The fault injection experiments though only consider dangerous failures. Basically, there are three types of fault injection techniques, where faults can be injected to memory and register [6-7]: hardware implemented, software implemented, and simulated fault injection. Among them, in Lee's study [5], only a limited hardware-implemented fault injection technique is used because some faults cannot be controlled when the full hardware-implemented fault injection technique is adopted. As a simple application, these steps are applied to a module in the integrated digital protection system (IDiPS) in a reactor protection system (RPS), which is a fully digitalized system developed in Korea [8]. Among 689 dangerous failures (out of a total of 1788 identified failures, the remaining 1099 being safe failures), 98.605% of them are detected. That is, the fault detection coverage of the applied fault-tolerant techniques in an IDiPS is 98.605%.

2.3 Further considerations on fault detection coverage quantification

Lee's study [5] focused on the fault detection coverage of the union of applied fault-tolerant techniques. To make a digital system more reliable though, the individual fault detection coverage of each technique needs to be investigated, as well as whether a specific fault is covered by another technique or not. If all faults can be covered through several techniques in multiple levels by modifying existing techniques and adopting new techniques, the reliability of the digital system can be drastically increased, as a fault can be detected by a higher-level technique if there is some problem with a technique at a lower level. This is the basic philosophy behind the defense in depth concept (redundancy, diversity, and independence) in the nuclear field [9]. As an effective approach for the topic described above, characteristics of the faults which are not detected by existing techniques should be examined.

3 Test-based approaches for software reliability quantification

Software is essential in digitalized I&C systems. To guarantee the overall safety of digitalized NPPs, the reliability of the software must be properly quantified. There are roughly three methods for software reliability quantification [10]. One is the software reliability growth model (SRGM), which estimates the increment of software reliability based on its fault removal during actual operation. This method however is not appropriate for safety-critical software because of its uncertainty of sufficient failure sets and very high sensitivity to rare faults [11]. The Bayesian network (BN) is another method that combines disparate information about the software. Reasonable BN development requires developer expertise, qualified model parameters, documented activities for the software development process, and a quantifying process for qualitative evidence [12-13]. As a result of these challenges, subsequent estimates may have large uncertainty, which is not acceptable for safety-critical software [10]. Therefore, BN should be complemented or verified by another method; in this context, the third method needs to be properly developed, which is a test-based method. The test-based method can be divided into the black-box test and the white-box test. For the reliability quantification of safety-critical software, the white-box test is superior. In this section, the limitations of the black-box test and related research based on the white-box test are reviewed.

3.1 Black-box approach for software reliability quantification

The black-box test considers software as a black box; *i.e.* it feeds inputs then examines whether outputs succeed or fail, but does not consider what happens inside of the software. To get the input sets for test execution, this method randomly samples input values from the operational profile distribution. Basically, a failure is revealed when specific input values trigger a certain faulty aspect of the software. In this sense, the averaged reliability based on the black-box method is valid only under the assumption that all the functions inside of the software are exercised through the test [14-16]. In actuality though, this assumption is difficult because of the uncertainty originating from its random sampling; expressly, during random sampling, the input values which will be selected in the future are unclear [17].

As a result of this uncertainty, the reliability quantification process of the black-box method can be based only on the number of tests executed and cannot be based on the coverage concept. Moreover, in this approach, further uncertainty arises from the ambiguity of what is a sufficient number of tests that needs to be considered. In this context, code characteristics (as in the white-box approach) should be utilized to eliminate the above uncertainties and to address the coverage concept.

3.2 White-box approach for software reliability quantification

To accurately quantify the reliability of software, testing should be executed in consideration of the test coverage concept. To discuss test coverage, all possible test cases first need to be clearly identified. Then, each test case should be addressed in real test execution; that is, rather than random sampling, a logical structure for the modification of the actual values of the parameters under software function needs to be developed. Basically, the white-box test considers the code characteristics inside of the software. Code characteristics, such as the assigned range of each variable and relations between variables, can be utilized to figure out the possible internal states of the software, which is formed by the combination of the stored values of each variable. By adopting a proper reference state variable (RSV) as a datum point, the possible values of other state variables can be scrutinized [18]. A variable indicating a process parameter would be a proper RSV for an RPS because most calculations and comparisons are conducted based on the process parameters.

In point of fact though, a test case is a combination of the internal state and inputs, so in order to identify all possible test cases not only code characteristics but also the input characteristics and relations between the internal state and inputs need to be considered. Kang *et al.* [19] proposed a systematic method for defining input characteristics based on the features of an analog to digital converter (ADC) and system dynamics. When an ADC has i bits of memory, the number of possible digital values is 2^i because the

analog signal should be converted into a certain range of digital values through the ADC. Under the specific resolution of an ADC, the possible input values of the next scan time depend on the scan interval (or scan time) and plant dynamics.

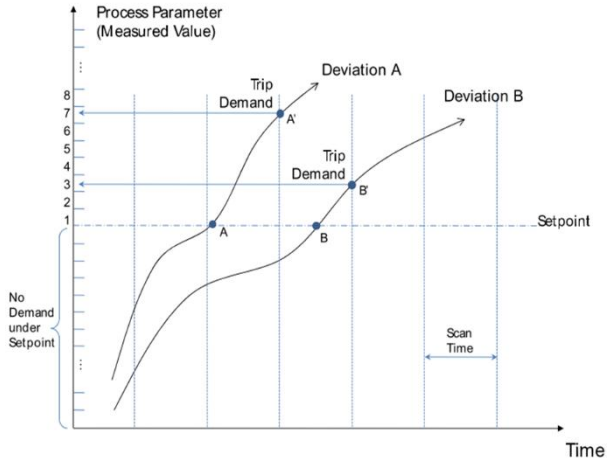

Fig. 2 Illustration of scan time and demand generation in consideration of input domain [19].

As an example, Fig. 2 illustrates possible inputs (here, the process parameter) in consideration of scan time and plant dynamics. For deviation A, the possible deviation of the process parameter (A') from the set point can move further away if scanning is performed sporadically and the process parameter is changed rapidly. In addition, scan timing is also important to decide the possible input domain as seen in the comparison between deviation A and B. Kang *et al.* demonstrate the feasibility of this approach by estimating the input profile of the pressurizer pressure in case of a loss of coolant accident (LOCA). This study provides a valuable insight to develop the input cases for a specific internal state.

When the possible internal states of software and the input domains for a specific internal state can be identified, the total number of required tests can be derived. Then, software tests can be conducted in the following simple manner: set a specific internal state, apply all possible input sets and check the correctness of the output, then set another internal state and apply the other possible input sets to the new internal state. Through this approach, the test cases that will not occur in actual use can be excluded while all other possible cases (representing the basis of the test coverage concept) can be tested. If all possible test cases are executed, it can be said that it is an exhaustive test. Even in the case though where there are some difficulties to conduct all possible test cases, some logical techniques, such as equivalent partitioning which divides the range of values of each parameter according to the expectation of the same output, can be adopted, and still preserve the test coverage concept.

4 Network communication risk

Utilization of network communication is very effective to reduce the number of complicated connections between various components and control modules in NPPs. Despite this, ecumenical research on comprehensive reliability assessments of safety-critical networks is still very rare. In fact, reliability quantification of any network focused on hardware alone is already an intricate task. Added consideration of software and network protocol makes the problem even more complex. While most research on network reliability is, on account of these difficulties, based on simulation or testing, Lee *et al.* [20] analyzed a network communication system of the engineered safety feature-component control system (ESF-CCS) in a systematic and comprehensive way. Therefore, Lee's study is introduced in this overview with the expectation that it could provide valuable guidance for the reliability quantification of safety-critical networks.

4.1 Identification of hazardous states and failure causes

The ESF-CCS employs a high reliability-safety data network (HR-SDN) for the transmission of safety-critical information from group controllers (GS) to loop controllers (LC) to accommodate the vast number of field components. The HR-SDN uses the Profibus-decentralized periphery (DP) protocol which is similar to that of the token bus protocol [21]. IEEE standard 802.4 specifies the operation mechanism of explicit token passing schemes to control access on a bus topology network [22]. There are four major processes in the Profibus-DP protocol: token frame reception, data frame transmission, data frame reception, and token frame passing [23]. When any of the above processes fail, the transmission of safety-critical information from GCs to LCs will fail, and the system can encounter hazardous states

corresponding to a failure of ESF signal generation initiation.

There are two types of failure causes in the Profibus-DP protocol: isolating errors which can be isolated to a given fault domain (a station, upstream neighbor, and wire between them) and non-isolating errors (lost frames, congestion, token errors, and frequency errors) [24]. In Lee's study [20], the isolating errors were treated as the main failure causes, and then these causes were categorized into hardware failure, software failure, and medium-related failure. Based on the specification [22], the hazardous states and their detailed causes are identified and listed in Table 1.

Table 1 Identified hazardous states and the corresponding causes of failure [20]

Hazardous States	Failure Causes
Token reception failure	-Failure of network interface module of station -Failure of receiver in network module of station -Failure of software function in network module of station -Token frame corruption caused by bit errors in medium
Data transmission failure	-Failure of network interface module of station -Failure of transmitter in network module of station -Failure of software function in network module of station
Data reception failure	-Failure of network interface module of station -Failure of receiver in network module of station -Failure of software function in network module of station -Data frame corruption caused by bit errors in network medium
Token passing failure	-Failure of network interface module of station -Failure of transmitter in network module of station -Failure of software function in network module of station

4.2 Quantification of network failure probability

The failure of the hardware or software of a network module may cause network failure. In addition, environmental interference in the medium may also cause faults in a token or data frame and result in network failure. These three factors should therefore be considered to estimate the risk of network communication.

The HR-SDN system is based on a safety-grade programmable logic controller (PLC), consisting of various modules including input, process, output, and network modules [25]. In Lee's study [20], the quantity and sub-level components of each module are investigated and the failure rates for each component are cited from proper references. Then, to estimate the hardware failure probability, the mean unavailability concept is adopted. The process for the mean unavailability calculation involves two periodic test intervals: a monthly manual test and an automatic self-diagnostic test assumed to be done every 50 milliseconds. In the sensitivity study, the important failure causes contributing to overall network failure for each case having different test intervals were isolated and analyzed. The dominant cause was hardware failure when the manual test interval is considered, whereas it was software failure when the self-diagnostic test interval is considered. Thus, a further study is needed to set the appropriate conditions for the test intervals to calculate mean unavailability.

To derive the software failure probability, a qualitative approach can be utilized that considers software complexity and the integrity of the verification and validation (V&V) process [26]. As an estimator for V&V integrity, software integrity level (SIL) is used. Since errors of the software implemented in GC and LC are recognized to occur infrequently but with critical consequences, the SIL of the software falls into 1 or 2 [27]. Then the complexity of the software is considered as low because it just focuses on the activation of safety-critical functions. Therefore, the software failure probability is assumed to range from 1.0E-04 to 1.0E-05, as shown in Table 2.

Table 2 Baseline failure probability estimates for various software conditions [26]

SIL	Complexity of the software		
	High	Medium	Low
0	1.0E-01	1.0E-02	1.0E-03
1	1.0E-02	1.0E-03	1.0E-04
2	1.0E-03	1.0E-04	1.0E-05
3	1.0E-04	1.0E-05	1.0E-06
4	1.0E-05	1.0E-06	1.0E-07

When data are transmitted over the transmission medium, errors may be introduced into the network module as a result of environmental interference. To quantify the probability of this risk, the operation

modes of safety-critical I&C systems need to be considered. Between the continuous and low-demand modes, most safety-critical instrumentation falls into the low-demand mode as operation is called for only in some abnormal states of NPPs. Thus, the probability of error occurrence in the medium can be treated as the probability of failure on demand. In terms of the probability of error introduction into the medium, the bit error rate (BER) can be used, which is the ratio of the number of bit errors in the transmitted bits to the total number of transmitted bits [28]. In general, BER is applicable for fiber-optic data systems, such as a Profibus-DP network, that transmit data over a transmission medium where environmental interference may cause corruption of the digital signal. In Lee's study [20], the estimated number of erroneous bits in each frame was treated to depend on the length of the token and data frames in the Profibus-DP protocol.

The application of a fault-tree analysis to safety-critical digital systems provides various advantages, including the reflection of multi-channel configuration and the identification of the critical factors in system safety. In this sense, the fault-tree method is suitable to analyze the ESF-CCS, as it has four redundant channels and each channel consists of three redundant GCs and doubly redundant LCs. Accordingly, as a case study, Lee [20] developed a fault-tree analysis of ESF-CCS signal failure in the containment spray actuation signal (CSAS). Based on the quantification results for each failure cause in four cases with different baseline software failure probabilities and periodic inspection intervals, it was found that network failure can contribute up to 1.88% of the probability of ESF-CCS signal failure for the CS pump considered in the case study.

7 Concluding remarks

At present, most non-safety NPP I&C systems are already digitalized and safety-critical functions are on the way of digitalization. Yet there are difficulties to ensure the safety of digitalized features, as an appropriate and comprehensive reliability quantification method for digitalized computer systems has still not been provided. In this context, recent noteworthy research related to three topics considered to be critical factors for reliability quantification was introduced here. Although the aforementioned studies still have some challenges to overcome for more practical implementations, they all provide worthy insights to set up a proper reliability quantification method for digitalized I&C systems.

So far, quite a lot of related research has been performed with valuable results accumulated. However, a general logical frame integrating all the factors related to the reliability quantification of digitalized I&C systems is still on its way of development. In consideration of conventional PSA modeling structure, fault tree would be one of most promising options. In reality the various factors composing digitalized I&C systems are not independent of each other but rather closely connected. Thus, from a macro point of view, a method that can integrate risk factors with different characteristics needs to be considered together with the micro approaches to address the challenges facing each factor.

Nomenclature

NPP	nuclear power plant
FMEA	failure mode and effects analysis
IDiPS	integrated digital protection system
RPS	reactor protection system
I&C	instrumentation and control
SRGM	software reliability growth model
RSV	reference state variable
ADC	analog to digital converter
LOCA	loss of coolant accident
ESF-CCS	engineered safety feature-component control system
HR-SDN	high reliability-safety data network
GS	group controller
LC	loop controller
DP	decentralized periphery
PLC	programmable logic controller
V&V	verification and validation
SIL	software integrity level
BER	bit error rate
CSAS	containment spray actuation signal

Acknowledgement

This work was supported by the Nuclear Research & Development Program of the National Research Foundation of Korea (NRF) funded by the Ministry

of Science, ICT & Future Planning (Grant Number: NRF-2015M2A8A4021648)

References

[1] CHAPIN, D., DUGAN, J. B., BRAND, D., CURTISS, J. DAMON, D., and *et al.*: Digital Instrumentation and Control Systems in Nuclear Power Plants, National Research Council: National Academy Press, 1997.

[2] ALDEMIR, T., STOVSY, J., KIRSHENBAUM, D., MANDELLI, P. BUCCI, L. A., and *et al.*: Dynamic Reliability Modeling of Digital Instrumentation and Control Systems for Nuclear Reactor Probabilistic Risk Assessments: U.S. Nuclear Regulatory Commission, 2007.

[3] KANG, H. G., and SUNG, T.: A Quantitative Study on Important Factors of the PSA of Safety-Critical Digital Systems, Nucl. Eng. Technol, 2001, 33: 596–604

[4] DUGAN, J.B., and TRIVEDI, K.S.: Coverage Modeling for Dependability Analysis of Fault-Tolerant Systems, IEEE Trans. Comput, 1989, 38(6): 775–787.

[5] LEE, S. J., CHOI, J. G., KANG, H. G., and JANG, S. C.: Reliability Assessment Method for NPP Digital I&C Systems Considering the Effect of Automatic Periodic Tests, Ann Nucl Energy, 2010, 37(11): 1527–1533.

[6] PINNA, T., BOCCACCINI, L. V., and SALAVY, J. F.: Failure Mode and Effect Analysis for the European Test Blanket Modules, Fusion Eng Des, 2008, 83(10-12): 1733–1737.

[7] HSUEH, M.-C., TSAI, T. K., and IYER, R. K.: Fault Injection Techniques and Tools, Computer, 1997, 30(April): 75–82.

[8] HUR, S., KIM, D. H., HWANG, I. K., LEE, C. K., and LEE, D. Y.: The Automatic Test Features of the IDiPS Reactor Protection System, In: KNS Spring Conference, Korea, 2007

[9] CEPCEK, S., DENISLAMOV, A., DOMENECH, H., HINTTALA, J., and *et al.*: IAEA Safety Standards Series: Maintenance, Surveillance and In-Service Inspection in Nuclear Power Plants: International Atomic Energy Agency. 2002.

[10] CHU, T., YUE, M., MARTINEZ-GURIDI, G., and LEHNER, J.: Review of Quantitative Software Reliability Methods: U.S. Nuclear Regulatory Commission, 2010

[11] KIM, M., JANG, S., and HA, J.: Possibilities and Limitations of Applying Software Reliability Growth Models to Safety-Critical Software, Nucl Eng Technol. 2007, 39:129–132

[12] FENTON, N., NEIL, M., MARSH, W., HEARTY, P., MARQUEZ, D., KRAUSE, P., and *et al.*: Predicting Software Defects in Varying Development Lifecycles Using Bayesian Nets, Inf Softw Technol. 2007, 49:32–43.

[13] FENTON, N., NEIL, M., and MARQUEZ, D.: Using Bayesian Networks to Predict Software Defects and Reliability, Proc Inst Mech Eng Part O J Risk Reliab. 2008, 222:701–712.

[14] MAY, J., HUGHES, G., and LUNN, A.: Reliability Estimation from Appropriate Testing of Plant Protection Software, Softw Eng J. 1995, 10:206–218.

[15] MAY, J., and LUNN, A. D.: A Model of Code Sharing for Estimating Software Failure on Demand Probabilities, IEEE Trans Softw Eng. 1995, 21:747–753

[16] MILLER, K., and MORELL, L.: Estimating the Probability of Failure When Testing Reveals No Failures, IEEE Trans Softw Eng. 1992,18:33–43

[17] KUBALL, S., and MAY, J.: A Discussion of Statistical Testing on a Safety-Related Application, Proc Inst Mech Eng Part O J Risk Reliab. 2007, 221:121–132

[18] SHIN, S. M., KIM, H. E., LEE, S. J., and KANG, H. G.: Finite Test Sets Development Method for Test Execution of Safety Critical Software. In: KNS 2014 autumn meeting. Pyeongchang, 2014

[19] KANG, H. G., LIM, H. G., LEE, H. J., KIM, M. C., and JANG, S. C.: Input-Profile-Based Software Failure Probability Quantification for Safety Signal Generation Systems, Reliab Eng Syst Saf, 2009, 94:1542–1546

[20] LEE, S. H., KIM, H. E., SON, K. S., SHIN, S. M., LEE, S. J., and KANG, H. G.: Reliability Modeling of Safety-Critical Network Communication in a Digitalized Nuclear Power Plant, Reliab Eng Syst Saf, 2015, 144: 285–295.

[21] WILLIG, A., and WOLISZ, A.: Ring stability of the PROFIBUS token-passing protocol over error-prone links, IEEE Trans. Ind. Electron, 2001, 48(5) 1025-1033.

[22] IEEE.: IEEE Standards for Local Area Networks: Token-Passing Bus Access Method and Physical Layer Specification, American National Standards Institute, IEEE, 1985.

[23] ELAHI, A.: Network Communications Technology, Southern Connecticut State University, Thomson Learning, 2001.

[24] HAUGDAHL, J. S.: Network Analysis and Troubleshooting, Addison-Wesley, 2000.

[25] KOO, S. R., and SEOUNG, P. H.: Software Design Specification and Analysis Technique (SDSAT) for the Development of Safety-Critical Systems Based on a Programmable Logic Controller (PLC). Reliab Eng Syst Saf, 2006, 91(6): 648–664.

[26] BACKSTROM, O., HOLMBERG, J., JOCKENHOEVEL-BARTTFELD, M., and TAURINES, A.: Quantification of Reactor Protection System Software Reliability Based on Indirect and Direct Evidence. In: Probabilistic Safety Assessment and Management, Hawaii, 2014

[27] IEEE Computer Society.: IEEE Standard for Software Verification and Validation, IEEE Computer Society, IEEE, 2005.

[28] JERUCHIM, C.: Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communication Systems, IEEE J Sel Areas Commun, 1984, 2(1):153–170.