Quantitative risk assessment of digitalized safety systems

Sung Min Shin[a], Seung Jun Lee[b], Sang Hun Lee[a], Hyun Gook Kang[a*]
[a] *Department of Nuclear and Quantum Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea*
[d] *Department of Nuclear Engineering, UNIST, 50 UNIST-gil, Ulsan 44919, Republic of Korea*
[*] *Corresponding author: hyungook@kaist.ac.kr*

1. Introduction

Over the past few decades, various digital systems have been supplanting the analog systems in nuclear power plants (NPP). A report published by the U.S. National Research Council indicates that appropriate methods for assessing 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 issue, the methodology for the probabilistic safety assessment (PSA) of digital I&C systems has been studied. However, there is still no widely accepted method [2]. Kang and Sung found three critical factors for safety assessment of digital systems: detection coverage of fault-tolerant techniques, software reliability quantification, and network communication risk [3]. In this paper, recent noteworthy approaches and challenging points for each of these factors are briefly introduced.

2. Detection coverage of fault tolerance techniques

2.1 Characteristics of fault tolerance techniques and importance of fault detection coverage

Fault tolerance is the capability of a system to work properly in spite of the existence of faults. 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 be detected by another one at a higher level. Figure 1 shows this conceptual structure of multiple fault-tolerant techniques.

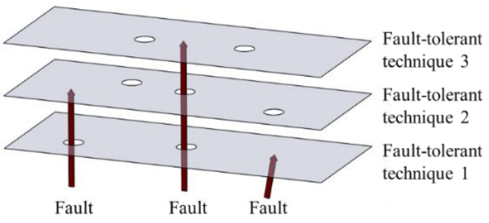
Fault-tolerant technique 3
Fault-tolerant technique 2
Fault-tolerant technique 1
Fault
Fault
Fault

Fig. 1 Faults and fault-tolerant techniques

In this sense, fault detection coverage, which is the ability to detect errors, is considered as one of the most crucial factors in the assessment process. 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.

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 the 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. 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, Lee et al. [5] took hardware-implemented fault injection technique to quantify the fault detection coverage in consideration of dangerous failures which effects causes an abnormal status of the system. As a simple application, Lee et al. applied this approach to a module in the integrated digital protection system (IDiPS) in a reactor protection system (RPS) [8]. Among 689 dangerous 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%.

Lee's study [5] focused on the fault detection coverage of the union of applied fault-tolerant techniques. However, 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, 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 of defense in depth concept in the nuclear field [9].

3. Test based approach 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]: software reliability growth model (SRGM), Bayesian network (BN), and test-based method. The SRGM is not appropriate for safety-critical software because of very high sensitivity to rare faults [11], and subsequent estimates of BN may have large uncertainty because of uncertainties in required evidence. In this context, BN should be complemented or verified by 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 based 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 [12-14]. 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 [15].

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 based 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, such as the assigned range of each variable and relations between variables, inside of the software. The code characteristics 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 [16].

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. [17] proposed a systematic method for defining input characteristics based on the features of an analog to digital converter (ADC) and system dynamics. 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.

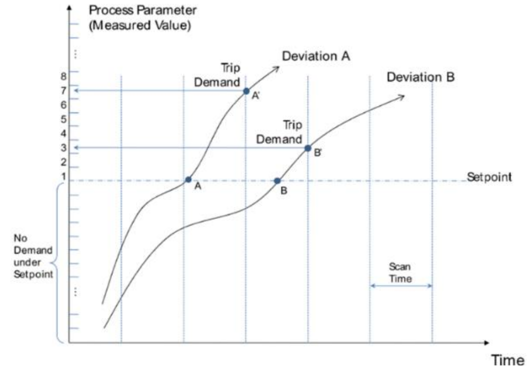

Fig. 2 Illustration of scan time and demand generation in consideration of input domain [19]

As an example, Figure 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 (representing the basis of the test coverage concept) can be derived. 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. Despite this, ecumenical research on comprehensive reliability assessments of safety-critical networks is still very rare because of complexity in correlation between hardware, software, and network protocol. While most research on network reliability is, on account of these difficulties, based on simulation or testing, Lee et al. [18] analyzed a network communication system of the engineered safety feature-component control system (ESF-CCS) in a comprehensive way. Therefore, Lee's study is introduced in this paper 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 [19]. IEEE standard 802.4 specifies the operation mechanism of explicit token passing schemes to control access on a bus topology network [20]. Lee et al. [18] identified the hazardous states and their detailed causes based on the specification [20] as shown in Table 1. In Lee's study, isolating errors which can isolated to a given fault domain (a station, upstream neighbor, and wire between them) were treated as the main failure causes, and then these causes were categorized into hardware failure, software failure, and medium-related failure.

Table 1 Identified hazardous states and the corresponding causes of failure [18]

Hazardous States	Failure Causes
Token reception failure	-Network interface module of station -Receiver in network module of station -Software function in network module of station -Token frame corruption caused by bit errors in medium
Data transmission failure	-Network interface module of station -Transmitter in network module of station -Software function in network module of station
Data reception failure	-Network interface module of station -Receiver in network module of station -Software function in network module of station -Data frame corruption caused by bit errors in network medium
Token passing failure	-Network interface module of station -Transmitter in network module of station -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 [21]. In Lee's study [18], 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 were different; 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 [22]. As an estimator for V&V integrity, software integrity level (SIL) is used. Based on the characteristics of the software implemented in GC and LC [22-23], the software failure probability is assumed to range from 1.0E-01 to 1.0E-05.

When data are transmitted over the transmission medium, errors may be introduced into the network module as a result of environmental interference. Therefore, this risk should be quantify properly. 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 [24]. In Lee's study [18], 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.

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 containment spray pump considered in the case study.

5. Concluding remarks

So far, quite a lot of related research has been performed with valuable results accumulated. However, although the fault-tree analysis is demonstrated in preceding studies, a general logical frame integrating all the factors related to the reliability quantification of digitalized I&C systems is still ambiguous. 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.

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., 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., 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., 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., 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., 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., 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., 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., 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., 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., LEHNER, J.: Review of Quantitative Software Reliability Methods: U.S. Nuclear Regulatory Commission, 2010

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

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

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

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

[15] KUBALL, S., 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

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

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

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

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

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

[21] KOO, S. R., 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.

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

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

[24] 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.