TEST BASED RELIABILITY QUANTIFICATION METHOD FOR A SAFETY CRITICAL SOFTWARE USING FINITE TEST SETS

Sung Min Shin, Sang Hun Lee and Hyun Gook Kang*
Department of Nuclear and Quantum Engineering
Korea Advanced Institute of Science and Technology
291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea
becomejade@kaist.ac.kr; titanic91@kaist.ac.kr; hyungook@kaist.ac.kr

Han Seong Son
The Department of Gaming
Joongbu University
201 Daehak-ro, Chubu-myeon, Geumsan-gun, Chungnam, 312-702, Republic of Korea
hsson@joongbu.ac.kr

Seung Jun Lee
Integrated Safety Assessment Division
Korea Atomic Energy Research Institute
Daedeok-daero 989-111, Yuseong, Daejeon, 305-353, Republic of Korea
sjlee@kaeri.re.kr

ABSTRACT

Software is currently used within nuclear power plants (NPPs) to digitalize many instrumentation and control (I&C) systems. To guarantee the safety of the NPP, the reliability of the software must be properly quantified. In this study, we propose a novel method for software reliability quantification. The method identifies and arranges possible internal states of the software that can occur in actual use. Based on a specific internal state, possible input sets (combination of single values of each input variable) are applied sequentially. In this process, the assigned range of each variable, correlation between variables, characteristics of analog-to-digital converter (ADC), and plant dynamics are considered to identify the possible states of each variable. The effectiveness of the proposed method is demonstrated via a case study for a trip logic in a reactor protection system (RPS). Compared with existing test-based methods, the proposed method can shorten test execution time and eliminate uncertainties derived from random sampling of input values from the operation profile. Moreover, this method can provide a number of test sets required for an exhaustive testing.

Key Words: Finite test set, Software reliability, Safety-critical software, Exhaustive testing

1 INTRODUCTION

Most instrumentation and control (I&C) systems in nuclear power plants (NPPs) are in the process of being digitalized in response to the extended features of digital systems and the difficulty in supplying analog components [1]. All digitalized I&C systems are equipped with dedicated software. Because a failure of the software can significantly affect the entire system [2, 3], the reliability of the software must be precisely quantified to guarantee the safety of the NPP. Many previous studies have investigated quantification schemes of safety-critical software reliability.

The software reliability growth model (SRGM) is widely used to assess the reliability of software by estimating the increment of reliability as a result of fault removal. Kim et al. [4] analyze the possibility of the application of the SRGM to safety-critical software. According to this reference, the SRGM is not applicable to safety-critical software because of the high sensitivity of the estimated number of faults to time-to-failure data and the uncertainty of the availability of sufficient software failure sets.

The Bayesian network (BN) is another method for software reliability assessment that can aggregate disparate information about software, such as software failure data and quality of software lifecycle activities, and include parameter uncertainties as a part of modeling [5]. Fenton et al. [6] develop a dynamic BN that can allow a BN to be tailored to different software development environments, such as a wide variety of life cycles. The same research group [7] also suggests a new approach to building BN models with 'continuous nodes', which was a traditional problem in the BN-based approach. Eom et al. [8] propose a V&V-based method using a BN to estimate the remaining faults in safety-critical software after the development life cycle is completed. However, developing a reasonable BN requires expertise of the BN developers, qualifications of experts to estimate model parameters, availability of thorough documentation of the software development activities, and a quantifying process of qualitative evidence. Due to the above challenges, the uncertainty in the resultant estimates may be very large. This large uncertainty may make it difficult to demonstrate the small failure probabilities associated with safety-critical software [5]. Therefore, the BN-based approach must be supplemented or verified by another method.

The test-based approach is another applicable method for the reliability assessment of safety-critical software. Some studies relevant to this approach have been conducted in the nuclear field [9-12]. These studies all insist that an input set for a software test must represent 'trajectories' (a series of successive values for the input variables of a program that occur during the operation of the software over time) by random sampling them from the input profile. Some variables of the software are stored in memory and changed according to the plant state and programmed logic. The combination of each value of the stored variables at a specific time forms an "internal state of software (ISoS)". When we recognize that the input values indicate the plant state, it can be said that the ISoS at a specific time depends on the past input history. A new input set forms a new ISoS. Even if the same input sets are entered into the software, different outputs could be generated because of different ISoSs. Thus, the above references insist on the use of the trajectory form of input. However, this approach also has the challenges of the uncertainty caused by random sampling, difficulties in addressing input coverage, ambiguity on the necessary length of a trajectory, a long test execution time, and the large number of test sets needed to demonstrate high reliability.

Kang et al. [13] present the finitude of the input domain for a digital system. A digital system treats inputs from instrumentation sensors in a discrete manner by using an analog-to-digital converter (ADC). If the ADC converts an analog input signal with k bits, the number of possible input digital values is 2^k. This value is the ADC resolution, which is associated with the increment between discrete values. Under the specific ADC resolution, the possible input domain depends on the scan interval and plant dynamics. The input domain will be large if scanning of the input value is performed sporadically and process parameters change rapidly. Although this reference does not consider the change of ISoS, it is important in that it suggests an approach for identifying the finiteness of the input domain.

In this study, we first investigated a method to identify the finiteness domain of the ISoS considering the assigned range of each variable, the resolution of each variable, and the correlation between variables. Possible domain input sets were investigated based on the identified specific ISoS. To establish the possible input sets, the assigned range of input variables was added to Kang's approach. When the finite domains for each ISoS and input set are identified and the ISoS is controllable, there is no need to form an unknown ISoS through the trajectory form of input. It is possible to test software using another type of input set, which is a combination of single values of each input variable. The total number of tests can be expressed mathematically based on this test scheme. The feasibility of the proposed method is demonstrated via a case study for a reactor protection system (RPS) trip logic.

Improvements in the proposed method relative to the trajectory input-based method are as follows:

- It eliminates the uncertainty problem in the random sampling of input values.

- It reduces the test execution time to a few milliseconds.

- It can provide a number of finite test sets for an exhaustive testing.

Digital I&C systems can be divided into two types [1], microprocessor-based systems and programmable logic device (PLD)-based systems. The PLD-based system provides more reliable performance than the microprocessor-based system because it can process the data in parallel and is tolerant to poor environmental conditions. Therefore, most safety digital systems are based on PLD. This study also focuses on the software for a PLD-based system in the nuclear field.

The output of software can be affected by H/W conditions as well, even if identical inputs are applied. However, the effects induced by the hardware conditions of running software, such as the aging of electrical elements, are not considered. This study only considers tests of the logical integrity of the software.

The remainder of this paper is organized as follows. In section 2, the procedure of identifying possible ISoSs is explained. Based on a specific ISoS, the method for identifying possible input sets is introduced in section 3. In section 4, a test process is described. The proposed method is demonstrated with a case study in section 5.

2 IDENTIFICATION OF POSSIBLE ISOSS

The main feature of a PLD-based digital system is indefinite and cyclic execution. It reads inputs, computes new states and compares input values with the stored values, and updates the outputs for each scan cycle. The computed and stored values will be used at the next scan time. For instance, in the nuclear field, values of process parameters, such as temperature, water level, and pressure, are obtained and compared with the trip setpoint to decide whether to generate a trip signal. The trip setpoint is a stored value at previous scan time, and it can be changed and stored based on input values at the current scan time for the next scan time. The important point is that there are some variables that are stored in memory. The stored values of each variable together form an ISoS. The variables that are computed and stored inside of the software at the last scan time are named state variables (SVs), and these SVs could be called for computation or comparison to the current scan time.

Although different past input sets are entered into the software, when the last values of each SV are the same, the ISoSs are treated as identical according to Equation (1). The trajectory form of input sets is applied to promote ISoS changes that may occur in actual situations. However, when the ISoS is controllable, there is no need to apply the trajectory form of input. In this context, the most challenging point is to identify all possible ISoSs.

$$ISoS_1 \cdots ISoS_{59}\, ISoS_{60} \cdots ISoS_{79}\, ISoS_{80} \cdots$$

$$ISoS_{60} = ISoS_{80} = ISoS_a,\ when\ ISoS_{60}\{SV_1, SV_2 \cdots SV_n\} = ISoS_{80}\ \{SV_1, SV_2 \cdots SV_n\} \qquad (1)$$

To investigate all possible ISoSs, the concept of a reference state variable (RSV) is adopted. The RSV can be a datum point to scrutinize possible states of other SVs. When the RSV is set to a specific value, possible states of other SVs will be restricted if they are correlated with the RSV, or all assigned states should be counted. By changing the value of the RSV within the assigned range, all possible states of each SV to the specific value of the RSV can be identified. The combined values of each SV will then be a possible ISoS. Through the consideration of correlation between variables, the number of tests will be reduced significantly by excluding some ISoSs that do not occur in actual use.

For this approach, the proper RSV should be selected, and then, all possible states of it have to be identified. A variable indicating a process parameter would be a proper RSV because in a PLD-based digital system, most calculations and comparisons refer to process parameters. Therefore, all possible ISoSs could be scrutinized based on the change in the process parameter value. The memory of the ADC must be known to identify all possible states of the process parameter. The analog signal from an instrumentation sensor will be converted into a discrete value through the ADC. The memory of the ADC plays an important role in this conversion process. If the ADC converts an analog input signal with k bits, the number of possible states will be 2^k. The discrete value here is typically called a count, and it is one of the possible states of this variable. The minimum count corresponds to the minimum analog signal (or physical process parameter), and the maximum count corresponds to the maximum analog signal (or physical process parameter).

3 IDENTIFICATION OF POSSIBLE INPUT SETS FOR A SPECIFIC ISOS

For the test execution, after setting a specific ISoS, possible input sets to the ISoS should be identified. Here, the input sets are also composed of relevant variables. The variables composing input sets are named input variables (IVs) and are obtained for the current scan time from outside of the software. As noted above, the input set for the test execution does not need to have a trajectory form but can be the combination of single values of each IV. In other words, each IV has only one value, as shown in Equation (2).

$$\begin{bmatrix} IV_1(1) & \cdots & IV_1(i) \\ \vdots & \ddots & \vdots \\ IV_n(1) & \cdots & IV_n(i) \end{bmatrix} \rightarrow \begin{bmatrix} IV_1(1) \\ \vdots \\ IV_n(1) \end{bmatrix} \tag{2}$$

Some IVs may be correlated to SVs. The possible states of the IVs can also be reduced according to this correlation. For this approach, one should determine which IVs correlate with which SVs. In the majority of cases, the input variables for a PLD-based digital system in the nuclear field are process parameters and manually generated signals for bypass or reset. The manually generated signals have no correlation with ISoSs, and can be generated at any time. However, the process parameters obtained for a current scan time are correlated with the process parameters of the previous scan time. More precisely, this correlation between previous and current process parameters comes from physical linearity. For instance, within a specific time gap, such as a scan interval, the pressure of the primary loop cannot exceed beyond a certain limit from the value of the previous scan time. Kang et al. [13] investigate this limit considering the characteristics of ADC and plant dynamics. The characteristics of the ADC are associated with the scan interval and the gap size of the discrete values, and the plant dynamics are associated with the transition slope of the process parameters. When a process parameter drastically changes and is sparsely scanned, the next obtained value can go further from the previous point. This reference utilizes simulation code to identify this possible drastic slope of transition. In addition to the factors considered in this reference, the assigned range and correlation to the SVs of IVs are considered in this study. At both ends of the possible values of RSV, the possible upper or lower states of IVs can be restricted by an assigned range as well as by ADC characteristics and plant dynamics. For instance, although the pressure of a primary loop can drop by 10 discrete values from the previous pressure point during a scan interval, it will be automatically readjusted to its minimum value when it goes below an assigned range. In a sense, the RSV plays a role as an initial state in Kang's approach. Based on this initial state, the characteristics of the ADC and plant dynamics can be considered along with the assigned range. This is the process used to identify the domain of possible states of correlated IVs.

$$D_{input} \text{ for } ISoS_n = D(plant\ dynamics \cap ADC\ characteristics \cap assigned\ range | ISoS_n) \tag{3}$$

4 DEVELOPMENT AND EXECUTION OF FINITE TEST SETS

When the domains for possible ISoSs and input sets for each ISoS are identified, tests can be conducted in the manner shown in Figure (1): set a specific ISoS - enter all possible input sets (combination of single values of each variable) to the specific ISoS - set another ISoS - enter all possible input set to this new ISoS in the same manner.

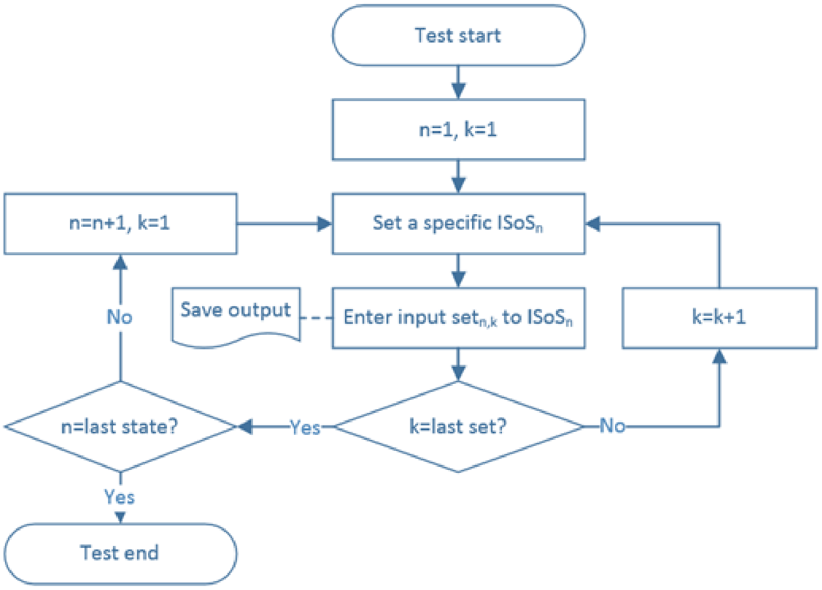

Figure 1. Test execution process

There are many advantages to conducting a test in this manner compared with a trajectory input-based approach. When an input trajectory is randomly selected from an operation profile, there is a potential limitation introduced by the reality of the operational profiles [10]. In the case of new software, this limitation becomes more severe because there may be no empirical data [12]. However, in the current approach, the operational profile does not appear necessary because the proposed approach examines all possible cases.

Basically, a software failure is revealed when specific input values trigger certain faulty part in software. In this context, there are two main uncertainties in the trajectory input-based approach, namely, where the faults lie in the software and which input values will be selected in the future [12]. Because of these uncertainties, a failure might not be reproducible [10]. The problems related to these uncertainties can be solved through the proposed method because it can control the internal state and have information about the values in input sets. Therefore, the failure surely can be reproduced even if there is uncertainty in the location of a fault.

Another significant problem with the trajectory input-based approach is the long time that each test set may take [10]. At worst, if a trajectory form focuses on a particular accident scenario which have a function that is not needed until a late stage, a test may take tens of hours. In contrast, the proposed test method will take only a few milliseconds for one test execution and does not need to consider various accident scenarios. The proposed method is simple and straightforward.

When correlations between variables are considered, the numbers of test sets for a certain value k of RSV and the total number of test sets can be expressed as shown in Equations (1) and (2), respectively. By considering this correlation, the number of test sets can be significantly reduced by excluding sets that will not occur in actual use.

$$N_k = \left[\left\{\prod_{i=1}^{n} N(SV_{no_i}) \times \prod_{i=1}^{m} N(SV_{co_i}|RSV_k)\right\} \times \left\{\prod_{i=1}^{l} N(IV_{in_i}) \times \prod_{i=1}^{k} N(IV_{co_i}|RSV_k)\right\}\right] \tag{4}$$

$$N_{test} = \sum_{k=min}^{max} N_k \tag{5}$$

where

N_k : Number of test sets when the RSV is k

$N(variable_i|RSV_k)$: Number of possible states of variable i under a certain value k of the RSV

SV_{no_i} : State variable that has no correlation

SV_{co_i} : State variable that has a correlation

IV_{no_i} : Input variable that has no correlation

IV_{co_i}: Input variable that has a correlation

N_{test} : Total number of test sets

5 CASE STUDY

The Korea Nuclear Instrumentation and Control System (KNICS) has recently developed a fully digitalized RPS based on a PLD [14]. As a digital system, this RPS is also equipped with dedicated software. The proposed method was applied to one of the trip logics in this RPS software to demonstrate its feasibility. There are 19 trip logics in the RPS. These logics can be divided into 3 categories according to their trip setpoint type [15]: “fixed trip setpoint”, “variable trip setpoint by manual reset”, and “variable trip setpoint by automatic rate-limiting”. The variable-type logics are more complex than the fixed logic because their setpoints can be changed. Among variable-type trip logics, only the pressurizer pressure low trip (PZR PR Lo Trip) logic has an operator bypass function, and it also has a reset delay timer by which the reset function is disabled for 10 s after resetting. Thus, this PZR PR Lo Trip logic was chosen because it was considered the most complicated logic.

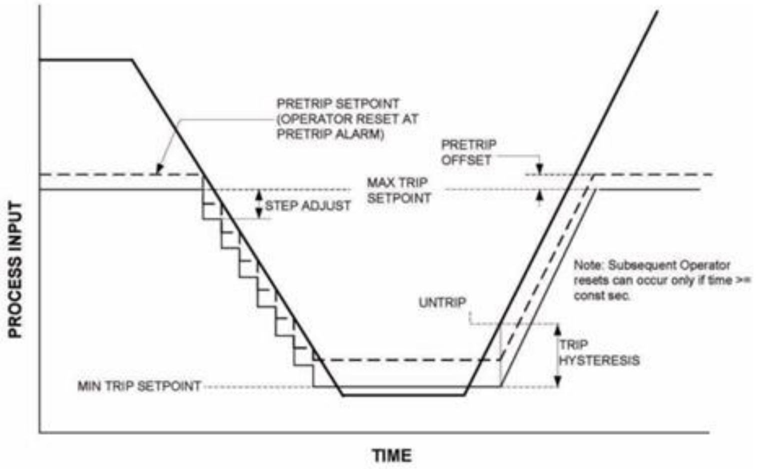
PRETRIP SETPOINT (OPERATOR RESET AT PRETRIP ALARM)
PRETRIP OFFSET
MAX TRIP SETPOINT
STEP ADJUST
PROCESS INPUT
Note: Subsequent Operator resets can occur only if time >= const sec.
UNTRIP
TRIP HYSTERESIS
MIN TRIP SETPOINT
TIME

Figure 2. Configuration of the variable trip setpoint by manual reset [11]

The “variable trip setpoint by manual reset” is described in depth in J. Choi and D. Lee's technical paper [1] by the name of “directional variable setpoint”. Fig. 2 shows the configuration of this trip logic. This logic will generate a trip signal when the system pressure drops to the trip setpoint level. However, the setpoint can be changed depending on the system pressure and operator reset, as shown in Fig. 2. When the system pressure rises away from the setpoint, the setpoint will chase the system pressure after 400 psi. However, when the system pressure falls back toward the setpoint, the setpoint will not be changed until the operator generates a reset signal. When the reset signal is generated, the trip setpoint will be dropped 400 psi from the system pressure at that time and cannot be changed again for 10 s. Bypass is permitted if the system pressure drops under 400 psi. The trip logic will be bypassed if the operator issues a bypass signal when bypass is permitted. Bypass permission will be removed automatically when the system pressure exceeds 500 psi again.

The variables inside the PZR PR Lo Trip logic were investigated. There are a total of 143 variables in this logic. When variables for mode selection, such as test or real mode, constant variables, and temporary variables that are automatically calculated based on input values between scan intervals, are excluded, the remaining variables can be summarized as 3 SVs and 5 IVs, as shown in Table I. Most of the variables are constants or temporary.

Table I. Summarized variables in PZR PR Lo trip logic

State variable	Input variable
Previous pressure	Current pressure
Trip setpoint	Bypass from MCR
Reset delay time	Bypass from RSR
	Reset from MCR
	Reset from RSR

Among the SVs, the previous pressure can be a RSV. The assigned physical range of this variable is 60 to 2,940 psi. To identify the number of possible states of this RSV, the characteristics of the ADC used for this RSV must be known. In the RPS of the optimized power reactor (OPR) 1000, a 12-bit ADC was used for system pressure [16]. This condition was referenced in this study. Therefore, the RSV can have 2^{12} states, and a state (count) corresponds to 0.703 psi.

The correlation of other variables to this RSV and the possible states of each variable based on this correlation should be analyzed to identify possible ISoSs. The correlation is summarized in Table II.

Table II. Correlation between the RSV and other variables

RSV	SV_{co}	IV_{co}	SV_{no}	IV_{no}
Previous pressure	Trip setpoint	Current pressure	Reset delay time	Bypass from MCR Bypass from RSR Reset from MCR Reset from RSR

As shown in Table 2, variables indicating the trip setpoint and current pressure are correlated to the previous pressure. The trip setpoint has an assigned physical range of 100 to 1,780 psi. Therefore, when the calculation logic results in a value above or below this range, it will be readjusted to 100 or 1,780 psi. As noted above, the value of this variable can be changed by the manual reset function. It will be dropped 400 psi from the value of the system pressure when the reset signal is generated, and it cannot be changed again for 10 s. In other words, the manual reset time (the system pressure at that time) will determine the value of the trip setpoint. The system pressure might increase or decrease after resetting. As the trip setpoint will chase the system pressure after 400 psi when the system pressure is raised and the trip setpoint will not change when the system pressure decreases, the value of the trip setpoint will always be located between the previous pressure and the previous pressure minus 400 psi and between 100 to 1,780 psi (assigned physical range).

The current pressure is also correlated with the previous pressure. The assigned range, ADC characteristics, and plant dynamics must be considered to identify the possible states of this variable. The assigned range of the current pressure is same as that of the previous pressure (100 to 1,780 psi). However, the possible deviation of falling or rising from the previous pressure is limited within successive scan intervals. Kang et al. [13] investigated the possible deviation of pressure falling from previous pressure according to several hole sizes of a loss of coolant accident (LOCA) scenario and scan intervals of the ADC. For this investigation, the thermal hydraulic system analysis code MARS (Multi-dimensional Analysis for Reactor Safety) was used. Among the conditions in this reference, we took the result of the largest hole size condition because it causes the largest pressure drop. We then took a scan interval of 50 ms because the software design specification (SDS) for KNICS RPS specifies that the processing time of the bistable processor (BP) should be shorter than 50 ms. When we consider these conditions, according to this reference, the current pressure can be dropped approximately 3.272 psi from the previous pressure between scan intervals, which corresponds to 5 counts. Chang et al. [17] investigated the pressure variation of the primary loop of OPR 1000 in the case of station block out (SBO) based on the MARS code. During pressure changes, the most drastic pressure increase was 0.5857 MPa in 1 s (=4.2 psi in 50 ms), which corresponds to 7 counts. These papers were referenced to identify possible falling or rising deviations of current pressure from the previous pressure.

There are 5 variables (1 SVs and 4 IV) that have no correlation with the previous pressure. The possible states of each variable also need to be identified. As noted above, the reset function is disabled for 10 s following a reset. In RPS, the elapsed time from the reset is calculated by a scanning counter. Therefore, there will be 201 states (10 s/50 ms + 1) for the reset delay time variable. The remaining variables (bypass from MCR or RSR, reset from MCR or RSR) are all Boolean-type variables with only 2 possible states for each variable.

When the previous pressure has a specific value k, the possible internal states of the PZR PR Lo Trip logic can be expressed by combining the possible states of the trip setpoint and the reset delay time, and for a specific ISoS, the possible input sets can be expressed by combining the possible states of the IVs (current pressure, bypass from MCR or RSR, reset from MCR or RSR). The number of test sets for this k value of RSV can be expressed by multiplying the possible number of ISoSs by input sets according to Equation (4). The total number of test sets can then be calculated by summing the number of test sets for each value of the current pressure from 0 to 4,096 (2^{12}) counts. From this calculation, we obtained a finite number of test sets, $4.5\text{x}10^{10}$.

Demonstration of this case study is expected to provide a better understanding of the proposed method for developing finite test sets. However, the required number of test sets is quite big to test all. Therefore, a lot of computing power is needed to be supported for test. Otherwise, the operational profile can be applied to set priorities between test sets. In this approach, the coverage of test sets can be discussed.

6 CONCLUSION

A simple and straightforward software test method was suggested. The method identifies the possible ISoS and the input sets for a specific ISoS considering the assigned range of each variable, correlation between variables, characteristics of the ADC, and plant dynamics. The total number of test sets required is then expressed by combining the identified possible ISoSs and input sets.

The suggested method can address the limitations in the existing trajectory input-based approach. By applying the method, the uncertainties related to random sampling from the operation profile can be eliminated, and the test execution time can be reduced drastically to a few milliseconds. Moreover, this method can provide the number of test sets that is required for an exhaustive testing.

The proposed method was demonstrated via a case study for the PZR PR Lo Trip logic in RPS. For the test of this trip logic, 4.5×10^{10} test sets were required. Because this number is quite large to test sequentially, the outside computing power should be supported for the exhaustive testing. Otherwise, priority between test sets can be set by considering the operational profile. When using this approach, the coverage of test sets can be treated based on the required number of tests for an exhaustive testing.

7 ACKNOWLEDGMENTS

This work was supported by Nuclear Research & Development Program of the National Research Foundation of Korea grant, funded by the Korean government, Ministry of Science, ICT & Future Planning (Grant Code: 2013M2A8A4025991).

8 REFERENCES

1. J.-G. Choi and D.-Y. Lee, “Development of RPS Trip Logic Based on PLD Technology,” *Nucl. Eng. Technol.*, **vol. 44**, pp. 697–708 (2012).
2. H. G. Kang and T. Sung, “An analysis of safety-critical digital systems for risk-informed design,” *Reliab. Eng. Syst. Saf.*, **vol. 78**, pp. 307–314 (2002).
3. H. Kang and S. Jang, “Fault-tree modeling for the signal generation failures of the engineered safety features in digitalized nuclear power plant,” *Saf. Reliab. Manag. Risk*, pp. 2431–2436 (2006).
4. M.-C. Kim, S.-C. Jang, and J. Ha, “Possibilities and limitation of applying software reliability growth models to safety-critical software,” *Nucl. Eng. Technol.*, **vol. 39**, pp. 129–132 (2007).
5. T. Chu, M. Yue, G. Martinez-Guridi, and J. Lehner, “Review of Quantitative Software Reliability Methods,” *Brookhaven Natl. Lab. Lett. Rep.* (2010).
6. N. Fenton, M. Neil, W. Marsh, P. Hearty, D. Marquez, P. Krause, and R. Mishra, “Predicting software defects in varying development lifecycles using Bayesian nets,” *Inf. Softw. Technol.*, **vol. 49**, pp. 32–43 (2007).
7. N. Fenton, M. Neil, and D. Marquez, “Using Bayesian networks to predict software defects and reliability,” *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.*, **vol. 222**, pp. 701–712 (2008).
8. H. Eom, G. Park, S. Jang, H. S. Son, and H. G. Kang, “V&V-based remaining fault estimation model for safety–critical software of a nuclear power plant,” *Ann. Nucl. Energy*, **vol. 51**, pp. 38–49 (2013).
9. J. May, G. Hughes, and A. Lunn, “Reliability estimation from appropriate testing of plant protection software,” *Softw. Eng. J.*, **vol. 10**, pp. 206–218 (1995).
10. T.-L. Chu, M. Yue, G. Martinez-Guridi, and J. Lehner, “Development of Quantitative Software Reliability Models for Digital Protection Systems of Nuclear Power Plants,” 2013.
11. S. Kuball and J. H. R. May, “A discussion of statistical testing on a safety-related application,” *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.*, **vol. 221**, pp. 121–132 (2007).
12. L. Strigini and B. Littlewood, “Guidelines for Statistical Testing,” *Tech. Rep.* (1997).
13. H. G. Kang, H. G. Lim, H. J. Lee, M. C. Kim, and S. C. Jang, “Input-profile-based software failure probability quantification for safety signal generation systems,” *Reliab. Eng. Syst. Saf.*, **vol. 94**, pp. 1542–1546 (2009).
14. K. Kwon and M. Lee, “Technical review on the localized digital instrumentation and control systems,” *Nucl. Eng. Technol.*, **vol. 41**, pp. 447–454 (2009).

15. G.-Y. Park, K.-Y. Koh, E.-Y. Jee, P.-H. Seong, K.-C. Kwon, and D.-H. Lee, “Fault Tree Analysis of Knics Rps Software,” *Nucl. Eng. Technol.*, **vol. 40**, pp. 397–408, (2008).

16. W. E. Company, “Ulchin nuclear power plant units 5 and 6: digital plant protection system technical manual,” (2002).

17. S. H. Chang, S. H. Kim, and J. Y. Choi, “Design of integrated passive safety system (IPSS) for ultimate passive safety of nuclear power plants,” *Nucl. Eng. Des.*, **vol. 260**, pp. 104–120, (2013).