Experimental Approach to Evaluate the Reliability of Digital I&C Systems in Nuclear Power Plants

Seung Jun Lee[a], Man Cheol Kim[b], and Wondea Jung[a]
[a] Korea Atomic Energy Research Institute, Daejeon, Korea
[b] Chung-ang University, Seoul, Korea

Abstract: Owing to the unique characteristics of digital instrumentation and control (I&C) systems, the reliability analysis of digital systems has become an important element of probabilistic safety assessments. In this work, an experimental approach to estimate the reliability of digital I&C systems is considered. A digitalized reactor protection system was analyzed in detail, and the system behavior was observed when a fault was injected into the system using a software-implemented fault injection technique. Based on the analysis of the experimental results, it is possible to not only evaluate the system reliability but also identify weak points of fault-tolerant techniques by identifying undetected faults. The results can be reflected in designs to improve the capability of fault-tolerant techniques.

Keywords: Digital I&C system, PSA, fault injection, fault-tolerant technique, failure detection coverage

1. INTRODUCTION

The probabilistic risk/safety assessment (PRA/PSA) has been widely used in the nuclear industry for licensing and identifying vulnerabilities to plant safety since 1975. PSA techniques are used to assess the relative effects of contributing events on system-level safety or reliability. They provide a unifying means of assessing physical faults, recovery processes, contributing effects, human actions, and other events that have a high degree of uncertainty [1,2].
Recently, instrumentation and control systems (I&C) in nuclear power plants (NPPs) have been changed into digitalized systems. Deterioration and an inadequate supply of components of analog I&C systems have caused inefficiency and high maintenance costs. Moreover, since the fast evolution of digital technology has made it possible to design more reliable functions for NPP safety, the transition from analog to digital has been accelerated. Owing to the unique characteristics of digital I&C systems, a reliability analysis of the digital systems has been introduced as one of the important issues in the PSA field [3,4].
The report published in 1997 by US National Research Council states that appropriate methods for assessing safety and reliability are key to establishing the acceptability of digital I&C systems in safety-critical plants such as NPPs [3]. The HSE's guide also pointed out the importance of the PSA for software-based digital applications as a demonstration of safety [5]. However, there is no widely accepted method for digital I&C PSAs. Conventional PSA techniques cannot adequately evaluate all features of digital systems. Failure coverage, common cause failures, and software reliability are the three most critical factors in the safety assessment of digital systems [6].
This work suggests an experimental approach to evaluate the reliability of digital I&C systems.

2. CHARACTERISTICS OF DIGITAL I&C SYSTEMS

Digital I&C systems are designed based on software and have unique characteristics utilizing software. The following should be considered in digital I&C system reliability evaluations:

- Failure coverage: Digital I&C systems have various fault-tolerant techniques for enhancing the system reliability. In the reliability evaluation, the fault-tolerant techniques and their failure coverage must be considered. A fault is a source that has the potential of generating failures. Fault-tolerance is the system's capability to help the system perform correctly the

specific required functions in spite of the presence of faults. In a fault tolerance evaluation, failure detection coverage is a crucial factor [7]. Failure detection coverage is a measure of the system's ability to perform failure detection, failure isolation, and failure recovery. For evaluating the failure detection coverage, it is important to exclude the duplicated effect of fault-tolerant techniques since various fault-tolerant techniques are implemented simultaneously at each level of the system hierarchy, such as component-level fault detection algorithms (e.g., memory checksum, watchdog timer for detecting microprocessor halt), board-level self-diagnostics (e.g., loop back check for input and output module), and system-level error detection mechanisms (e.g., automatic periodic test, state comparison algorithm of redundant modules). In addition, a different inspection period and range of each fault-tolerant technique should be considered [8].

- Common cause failure (CCF): The issues related to a system are the risk concentration and diversity (including CCF), the failure coverage of a self/peer monitoring, the effectiveness of an automated periodic system testing, and the network communication failures. The use of a single microprocessor module for multiple safety-critical functions will cause a severe concentration of risk in a single microprocessor. Safety-critical applications have adopted a conservative design strategy, based on functional redundancies. However, the software programs of these functions are executed by one microprocessor sequentially. Therefore, the level of redundant design of digital systems is usually higher than those of conventional mechanical systems. This higher redundancy will clearly reduce the risk from a single component failure, but raise the severity of CCF consequence. This higher level of redundancy exponentially increases the number of CCF events modeled in a fault tree, if conventional CCF modeling methods are applied. In some nuclear power plants, there are four signal processing channels for the safety parameters, and each channel consists of two or four microprocessor modules for the same function. For example, in the RPS of the OPR-1000 plant, there are 16 processors that do the identical function of local coincidence logic. In this case, the system model will have 65519 events for representing the CCFs of the local coincidence logic processors [3].

- Software reliability: The prediction of software reliability using a conventional model is generally much harder than for hardware reliability. It is notable that there has been a lot of discussion among software engineering researchers about whether a software failure can be treated in a probabilistic manner. Software faults are design faults by definition. That is, software is deterministic and its failure cannot be represented by a 'failure probability'. However, software can be treated based on a probabilistic method because of the randomness of the input sequences [3].

In this work, a digitalized reactor protection system (RPS) was tested for evaluating its reliability using one of the fault injection techniques. A fault injection is a technique for validating the reliability by observing the system behavior when a fault is injected. It consists of the accomplishment of controlled experiments where the observation of the system's behavior in the presence of faults is induced explicitly. The target system is tested without decomposition, thus problems in the system such as CCF or software flaws are reflected in the test results. That means the fault injection method threatens not the components of a system, but the whole system as it is, and the experiment results include all possible effects of problems existing inside the system. A limited software-implemented fault injection technique in which faults can be injected into memory and register was used based on an assumption of that all faults in a system are reflected on the faults in the memory and register. To reduce the necessary fault injection experiments and obtain reliable results, the memory map of the target software was analyzed. An unnecessary fault injection can be eliminated and the importance of specific memory area can be identified based on the analyzed memory map.

3. FAULT INJECTION EXPERIMENTS

3.1. Target Digital Reactor Protection System

We propose an experimental approach to evaluate the reliability of digital I&C systems. For a more realistic evaluation, the prototypes of digital I&C systems that have been adopted in a real digitalized NPP were used for the experiment. The target digital I&C system is the Integrated Digital Protection System (IDiPS) Reactor Protection System (RPS), which was developed in Korea [9,10] during the Korea Nuclear Instrumentation and Control System (KNICS) research and development project. The IDiPS RPS has four independent channels, where each channel consists of bistable processors (BPs), coincidence processors (CPs), an automatic test and interface processor (ATIP), a cabinet operator module (COM), and other hardware components [11].

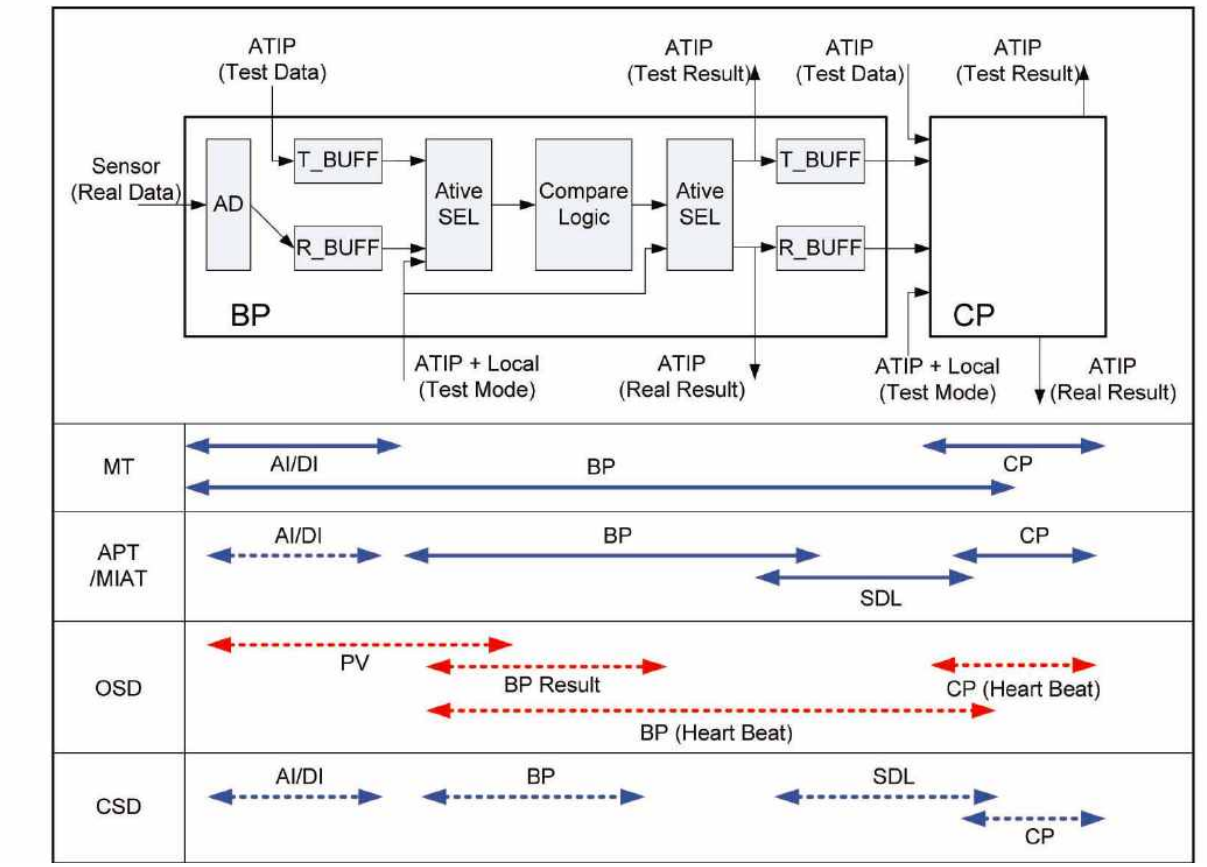

Figure 1: Fault-tolerant techniques in the IDiPS RPS [11]

IDiPS RPS tests can be classified into two categories: active tests and passive tests. Figure 2 shows four types of tests that have different types of coverage and periods [11].

- Active tests consist of automatic periodic tests (APTs), manual initiated automatic tests (MIATs), and manual tests (MTs) [9]. An APT is periodically initiated by the ATIP without any human intervention. An MIAT is almost the same as an APT except for the operator initiation and tested trip parameter selection. An MT is generally performed once per month.
- A passive test partially checks the system's integrity. This test consists of component self-diagnostics (CSD) and online status diagnostics (OSD).

In our work, the BP of the IDiPS-RPS was selected as a target system. Among the failure detection functions of the target system, three were considered: OSD, CSD, and APT.

3.2. Fault Injection Techniques

We used a software-implemented fault injection technique in which faults can be injected only into the memory [12,13]. Our fault injection experiment was conducted based on the assumption that all faults in a system are reflected in the faults in the memory because a fault should affect the memory related to the calculation process or variables and cause a wrong output. A fault of any component in a system may have an effect on the calculation process, reading input variables, generating output variables, and so on. A wrong calculation, program halt, variable changes, or wrong execution path may be caused by the fault. Conversely, the fault may have no effect on the output. If a fault does not have any effect on the output, then it is impossible to detect the fault because there are no observable consequences from the fault. If a variable related to the system output is changed by an inappropriate value for the current situation, then the fault may be detectable [14].
The fault injection experiment was performed for a system, not for a single component of the system. The different inspection period and range problem is not available because the behavior of the system against the injected fault was observed.
A fault injection experiment was performed based on the following three steps. First, fault types were identified according to the effects of injected faults. Based on the fault types, the failure detection coverage was defined. Second, a memory map of the target system was analyzed to perform efficient experiments. Unnecessary experiments were eliminated to reduce the number of experiments required. Finally, fault injection experiments were performed, and the results were analyzed.

3.3. Definition of Failure Detection Coverage

Faults in digital I&C systems are categorized into seven types according to their consequence and detection potential, as shown in Table 1.

Table 1: Categorization of faults into seven types

	Changed and used			Unused or unchanged
	Correct output	Wrong output	No output	
Detected	A	C	E	G
Undetected	B	D	F	

- Correct output (Fault types A and B):
  - Even when a bit is changed by a fault, and the changed bit is used to generate a system output, there may not be any effect on the output because the changed bit is not directly related to the output generation. For example, a stuck-at-1 fault changes “variable A” from 16 (binary: 10000) to 24 (binary: 11000). In this case, if the set point for “variable A” is 10, then the output is not changed, because both 16 and 24 are greater than the set point. This type fault is categorized as a safe fault.

- Wrong output (fault types C and D):
  - The bit changed by a fault may cause a wrong system output. For example, “variable A” has a value of 16 (binary: 10000), and the set point is 10. If the highest bit of “variable A” is changed by a stuck-at-0 fault, then “variable A” becomes 0 (binary: 00000) and a wrong output is generated.

- No output (fault types E and F):
  - The bit changed by a fault may cause a program halt or infinite loop, and thus the program does not generate an output. In this case, nothing is written on bits for the output, and the previous output is not updated.

- Unused or unchanged (fault type G):
  - A memory area is not assigned to any program code or variables. Even though some memory area is assigned and used, there will be no effect on the output unless a fault

changes a memory bit. For instance, if a stuck-at-0 fault is injected on a bit that was already 0, then nothing is changed. These unused or unchanged bits do not have any effect on the output generation, and it is impossible to detect such faults.

If a system works correctly despite the presence of a fault, the fault is called a "safe fault." A "correct output" (fault types A and B) and "unused or unchanged" (fault type G) fault types are classified as a "safe fault." Even if a malicious fault causes a "wrong output" or "no output" (fault types C, D, E, and F), if it is detected, the system will remain in a safe state. Such detectable malicious faults (fault types C and E) are also classified as a "safe fault" in terms of safety. If a malicious fault is not detected, the fault is classified as an "unsafe fault." The fault types are categorized as shown below:

- No-effect faults: A, B, G
- Malicious faults: C, D, E, F
- Safe faults: A, B, C, E, G
- Unsafe faults: D, F

We define the failure detection coverage as the probability of detecting malicious faults. The equation of failure detection coverage is defined as

$$(C + E) / (C + D + E + F) \quad (1)$$

3.3. Fault Injection Experiments

We performed fault injection experiments on the memory area of the BP application. Faults were injected into the memory of the BP application using the Code Composer tool [15], and an automatic fault injection program was developed for the experiment. Figure 2 shows the environment of the fault injection experiment. Two types of memory faults, stuck-at-0 and stuck-at-1, were considered because a memory bit has a binary value.

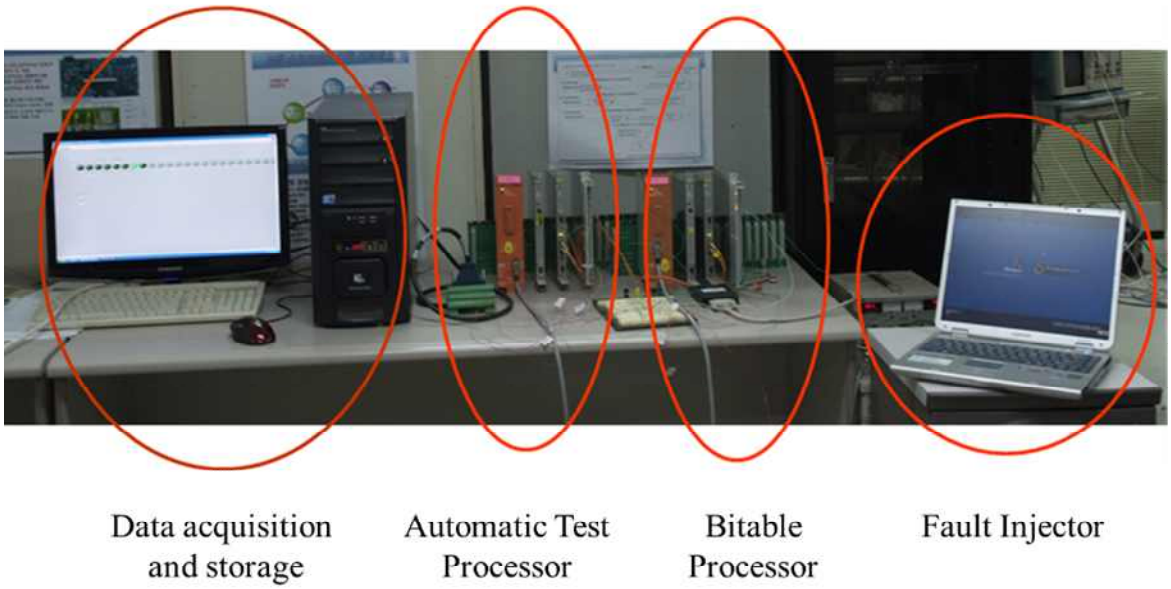

Figure 2: The environment for the fault injection experiment [11]

Because this experiment was for a feasibility study, the fault injection experiments were performed under limited conditions as follows in order to reduce the experimental time.

- A limited memory area was examined. A fault injection experiment for every single bit requires a large amount of time because of the large memory size, and each fault injection experiment takes approximately 1 min. For example, a total of approximately 8 million experiments are necessary just for the memory of the BP OS code. Moreover, the memory size of the BP application is much greater than that of the BP OS. Therefore, fault injections were performed on 3% of used memory area, and only two bits of each assembler line (the first and last bits) were examined. Usually, the first and last bits have a more significant effect than the other bits, and thus this limited condition may result in a more conservative output.

- The environment for the fault injection experiments is not exactly the same as the actual operating environment. The fault injection conditions differ from plant operating conditions even though actual digital I&C systems are examined, because the fault injection environment is implemented using only BP and ATIP. If other components are connected, then different behaviours can be observed. However, in terms of failure detection, it is expected that the results will differ little from those of the actual operating environment.

3.4. Experiment Results

A total of 55,752 fault injection experiments were then performed excluding the unused memory area, and the following observations were made.

- Faults resulting in no effect (fault types A, B, and G): (90.77% of injected faults)
- Faults resulting in no trip (fault types C, D, E, and F): 5,144 (9.23% of injected faults)
- Detected faults (fault types C and E): 5,028 (9.02% of injected faults)
- Undetected faults (fault types D and F): 116 (0.21% of injected faults)

Among the faults that caused a trip signal generation failure (C + E + D + F), the undetected faults (D + F) occupied 2.26%. Therefore, the failure detection coverage of the target system was 97.74%, based on Equation. 1.

3.5. System Reliability

The failure probability of an analog I&C system is calculated with the failure probabilities of the components. If an analog system consists of four relays and a failure of any relay causes a system malfunction, then the system failure probability is *p(relay failure probability)* $\times$ *4*. However, since digital I&C systems consist of hardware and software, software failure probability should be considered in addition to the hardware failure probability. Moreover, in spite of a system failure, a system malfunction is prevented if the failure is detected. The equation for the failure probability of a digital I&C system is as follows:

$$p(\text{digital I\&C system failure}) = (p(\text{HW failure}) + p(\text{SW failure})) * (1-p(\text{failure detection})) \quad (2)$$

Systems applied in NPPs are highly reliable and examined through strict validation/verification processes. In fact, no failure was observed when a fault was not injected in the experiments. Since only systems that do not have any flaw can be adopted in NPPs, it is not possible to estimate the system failure probability through this experiment. Usually, the failure probability of a component in analog I&C systems is about 1E-5 – 1E-6. If it is assumed that the failure probability of a digital RPS including hardware and software failure probability is 1E-5 and the failure detection coverage is 90%, then the system failure probability is 1E-6. If the failure detection coverage is 99%, then the system failure probability reduces to 1E-7. The failure detection coverage of fault-tolerant techniques is a very important factor to enhance the reliability.

4. CONCLUSION

The unique characteristics of digital I&C systems should be considered to estimate the reliability of digital I&C systems. In the present work, the reliability of digital I&C systems was estimated through fault injection experiments. A software-implemented fault injection technique in which faults are injected into the memory was used based on the assumption that all faults in a system are reflected in the faults in the memory. The fault injection experiment was performed based on the following three steps. First, fault types were identified according to the effects of the injected faults. Based on the fault types, the failure detection coverage was defined. Second, the memory map of the target system was analyzed to perform efficient experiments. Unnecessary experiments were eliminated to reduce the required number of experiments. Finally, fault injection experiments were performed, and the results were analyzed. For a feasibility study, a limited number of fault injections were performed using two digital I&C components. Based on the experimental results and analyzed memory map, the number of faults for each fault type was estimated.
Based on the experiment result analysis, it is possible not only to evaluate the reliability of digital I&C systems but also to point out the weakness of fault-tolerant techniques by identifying the undetected faults. The result can be reflected to the design to improve the capability of fault-tolerant techniques.

Acknowledgements

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: 2012M2A8A4025991).

References

[1] S. Authen and J. Holmberg. "*Reliability Analysis of Digital Systems in a Probabilistic Risk Analysis for Nuclear Power Plants*". Nuclear Engineering and Technology. 44, pp. 471-482, (2012).
[2] T. Aldemir, et al. "*Dynamic Reliability Modeling of Digital Instrumentation and Control Systems for Nuclear Reactor Probabilistic Risk Assessments*". NUREG/CR-6942, (2007).
[3] H. G. Kang, et al. "*An overview of risk quantification issues of digitalized nuclear power plants using static fault tree*". Nuclear Engineering Technology. 41, pp.849-858. (2009).
[4] M. Douglas, et al. "*Digital Instrumentation and Control Systems in Nuclear Power Plants*", National Academy Press, Washington, D.C, (1997).
[5] HSE, "*The use of computers in safety-critical applications"*, London, HSE Books, 1998.
[6] H. G. Kang and T. Sung. "*An analysis of safety-critical digital systems for risk-informed design*". Reliability Engineering and System Safety. 78, pp. 307-14, (2002).
[7] J. B. Dugan and K. S. Trivedi. "*Coverage Modeling for Dependability Analysis of Fault-Tolerant Systems*. IEEE Transactions on Computer. 38(6), pp.775-787, (1989).
[8] S. J. Lee, et al. "*Reliability assessment method for NPP digital I&C systems considering the effect of automatic periodic tests*", Annals of Nuclear Energy. 37, pp.1527-1533, (2010).
[9] K. C. Kwon and M. S. Lee. "*Technical Review on the Localized Digital Instrumentation and Control Systems*". Nuclear Engineering and Technology. 41, pp. 447-454, (2009).
[10] S. Hur, D. H. Kim, I. K. Hwang. "*A New Automatic Periodic Test Method for the Digital Reactor Protection System*". NPIC&HMIT, Knoxville, Tennessee, USA, (2009).
[11] J. G. Choi, et al. "*Fault Detection Coverage Quantification of Automatic Test Functions of Digital I&C System in NPPs*", . Nuclear Engineering and Technology. 44, pp. 421-428, (2012).
[12] S. J. Kim, et al. "*A method for evaluating fault coverage using simulated fault injection for digitalized systems in nuclear power plants*". Reliability Engineering and System Safety. 91, pp. 614-623, (2006).
[13] M. Hsueh, T. K. Tsai, R. K. Iyer. "*Fault injection techniques and tools*". IEEE Transaction on Computer. 30, pp. 75-82, (1997).

[14] J. S. Lee, et al. "*Evaluation of error detection coverage and fault-tolerance of digital plant protection system in nuclear power plants*". Annals of Nuclear Energy. 33, pp. 544-554, (2006).
[15] Texas Instruments, 1994. Code Composer, User's Guide.