Proceedings of the Korean Nuclear Society Autumn Meeting
Taejon, Korea, October 2000

AN INTEGRATED MODEL FOR RELIABILITY ESTIMATION OF DIGITAL NUCLEAR PROTECTION SYSTEM BASED ON FAULT TREE AND SOFTWARE CONTROL FLOW METHODOLOGIES

Man Cheol Kim and Poong Hyun Seong

Korea Advanced Institute of Science and Technology
373-1 Kusong-dong, Yusong-gu
Taejon 305-701, Korea

Abstract

In the nuclear industry, the difficulty of proving the reliabilities of digital systems prohibits the widespread use of digital systems in various nuclear application such as plant protection system. Even though there exist a few models which are used to estimate the reliabilities of digital systems, we develop a new integrated model which is more realistic than the existing models. We divide the process of estimating the reliability of a digital system into two phases, a high-level phase and a low-level phase, and the boundary of two phases is the reliabilities of subsystems. We apply software control flow method to the low-level phase and fault tree analysis to the high-level phase. The application of the model to dynamic safety system(DSS) shows that the estimated reliability of the system is quite reasonable and realistic.

I. Introduction

Due to many shortcomings of existing various analog systems, analog systems are being replaced with digital systems. But in spite of many advantages of a digital system compared to an analog system, the difficulty of proving the reliability of the digital system prohibits the widespread use of the digital system in the nuclear industry due to licensing problem. Besides, the reliability of a digital system is invaluable information for the maintenance of the system. This is the reason we have interest in establishing a method for estimating the reliability of a digital system.

There exists a few models for the purpose of estimating the reliability of a digital system. Goel and Soenjoto[1] developed a model for a hardware-software system and later the more generalized model was suggested by Sumita and Masuda[2]. Welke et al[3] developed a model which is applicable to more complicated situations. Vemuri and Dugan[4] suggested a reliability analysis method using dynamic fault tree analysis. Generally, these models make assumptions on the failure rates of subsystems which consists of hardware and software components. Therefore, it can be said that the interest of these models are high-level system reliability which focuses on the calculation of the reliability of a system based on the reliabilities of its constituting subsystems.

Recently, Choi and Seong[5] suggested a somewhat different model for estimating the reliability of a digital system which focuses on the estimation of the reliability of a subsystem based on the reliabilities of its hardware and software components. The interest of the model is low-level system reliability.

The models for estimating the high-level system reliability are usually strong at calculating the reliability of a complex system, but they are not so good at estimating the reliabilities of its subsystems where the interaction between their hardware and software components are

important. The model for estimating the low-level system reliability is strong at estimating the reliabilities of subsystems with consideration of hardware and software interaction. But it is not very good at calculating the reliability of a system with complex structure.

Based on these model comparison results, we develop a new model which integrates the advantages of each group of models. We separate the process of estimating the reliability of a digital system into two phases, low-level phase and high-level phase. In separating the process, the boundary of two phases is the reliabilities of subsystems. In other words, we first estimate the reliabilities of subsystems using a low-level system reliability model, and then calculate the reliability of the system which might have quite complex structure. After considering the advantages and disadvantages of each model, we adopt the hierarchical approach suggested by Choi and Seong[5] for low-level reliability estimation and fault tree analysis for high-level reliability estimation.

Other parts of this article describe the model we suggest and an application of our model to a digitalized nuclear power plant protection system, Dynamic Safety System(DSS). In section II, we briefly describe the low-level phase of estimating system reliability. In section III, we describe the high-level phase of estimating system reliability using fault tree analysis. In section IV, an application of our model to Dynamic Safety System is presented. And finally in section V, conclusions and further work is summarized.

II. Low-Level Phase

In low-level phase i.e. in estimating the reliabilities of subsystems, we adopt the hierarchical approach suggested by Choi and Seong[5]. The originality of this approach is its

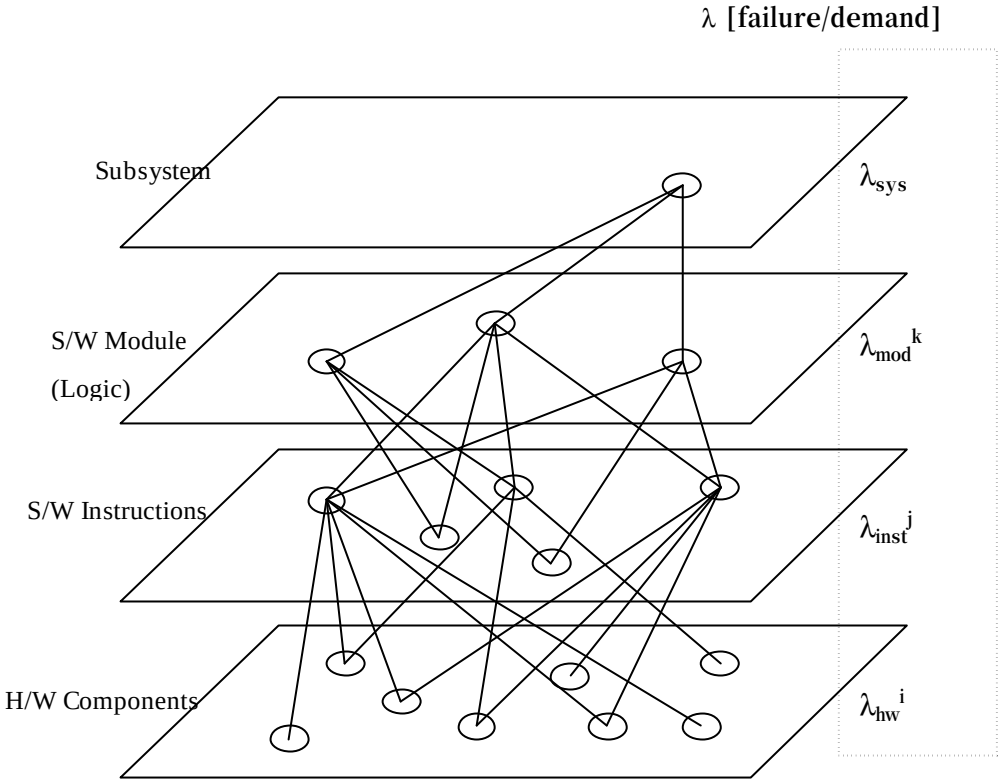

Fig.1 Function View of a Subsystem at Board Level [5]

recognition of the dominance of transient faults in hardware components and its proper modeling of transient-fault-induced-failures. The approach assumes that not every fault leads the system to failure, as opposed to most other models does.

Fault Classes

A hardware component may have faults induced by various causes. Some faults may be some physical defect or damage of a part in a hardware component, so it cannot be recovered. These kinds of faults are called "permanent faults". Most reliability models only consider this kind of faults, so if a fault occurs in a hardware component, the hardware component is thought to be out-of-service. Therefore, in calculating the reliability of a subsystem, if there are no redundancies in the subsystem, the failure rate of the subsystem is the sum of the failure rates of its hardware components and software components. It is worth to note that in this case the interaction between the hardware components and software components is not considered, and their failures are thought to be independent to each other.

However, all faults does not belong to "permanent faults" classes. Some faults may be only a sudden change of the state of a part in a hardware component. This kind of faults can be recovered or ignored after a refresh is performed or an overwriting of a new state is performed. In other words, if overwriting of the state occurs before the wrong state is used, system failure would not occur. This kind of faults usually appears and exists for a very short time and disappears, they are called "transient faults". According to Sun-2 file server system failure data with 21 workstation-years operation experience, it was found that "transient faults" are 10 times more often than "permanent faults"[6]. Because transient-fault-induced-failures are situation dependent, in estimating the reliability of a system the interaction between hardware components and software components must be taken into account.

Model Description

Fig. 1 shows the functional view of a subsystem at board level. In the fig. 1, a subsystem is composed of 3 software modules. Each software module consists of software instructions that a specific microprocessor provides. And each software instruction uses hardware components such as microprocessors, Input/Output ports, and memories. When considering only "permanent faults", the failure rate of the subsystem will be as follows:

$$l^{p}_{sys} = \sum l^{p}_{H/W} + \sum l^{p}_{S/W} \qquad (1)$$

where

l^{p}_{sys} : the permanent-faults-induced-failure rate

$l^{p}_{H/W}$: the permanent fault rate of a hardware

$l^{p}_{S/W}$: the permanent fault rate of a software

As stated earlier, when considering "transient faults", the interaction between hardware components and software components must be taken into account. When a hardware component is in idle state, i.e. the hardware component is not used by a software instruction, a transient fault of the hardware component does not cause the failure of the subsystem. Failure of the subsystem occurs when a hardware component used by a software instruction encounters a transient fault. It means :

$$l^{j}_{inst} = m^{j}_{inst} \sum l^{t}_{H/W} \qquad (2)$$

where

l^{j}_{inst} : the failure rate of jth software instruction

m^{j}_{inst} : the clocktime needed to process jth instruction

$l^{t}_{H/W}$: the transient fault rate of a hardware

The failure rate of a software module is as follows:

$$l^{k}_{\mathrm{mod}} = \sum_{j} p_i m^{k,i}_{inst} l^{j}_{inst} \quad (3)$$

where

l^{k}_{mod} : the failure rate of *k*th software module

p_j : the software branch probability

$m^{k,i}_{inst}$: the total number of *j*th instruction usage

The transient-fault-induced-failure rate of the subsystem is represented as follows:

$$l^{t}_{sys} = \sum_{k} l^{k}_{\mathrm{mod}} \quad (4)$$

where

l^{p}_{sys} : the transient-faults-induced-failure rate

The failure rate of a subsystem can be calculated by adding the permanent-fault-induced-failure rate and the transient-fault-induced-failure rate.

$$l^{p}_{sys} = \sum l^{p}_{H/W} + \sum l^{p}_{S/W} + \sum_{k} l^{k}_{\mathrm{mod}} \quad (5)$$

III. High-Level Phase

Once the failure rates of subsystems are estimated, it is much easier to calculate the reliability of the whole system. There are a few methods applicable for this phase, reliability block diagram(RBD), Markov chain, fault tree analysis(FTA) and Monte Carlo simulation. Each of these methods have their own advantages and disadvantages.

Reliability Block Diagram (RBD)

Reliability block diagram is good at modeling a simple-structured system. After building a proper diagram, the reliability of the system as a function of time can be calculated analytically. Using cutset generation algorithms, critical components of the system are easily identified. But for a system with a quite complex structure, building reliability block diagram becomes a real challenge. Even if the diagram is successfully built, the analytic solution obtained from the diagram tends to be too complex to be practically useful. In building a diagram, there exists some limitations in translating a real system into a diagram, such as external dependencies.

Markov chain

Markov chain is good at modeling a simple-structured system in which each subsystem can be in several different states. When the structure of a system under study is quite simple, it is possible to calculate analytic solution for the reliability of the system. Besides, other information concerning to the behavior of the system and the transition between various states in the system can be calculated in an analytic manner. However, for a complex system, the size of Markov chain for the system explodes exponentially, the chain becomes too huge to be manageable.

Fault Tree Analysis (FTA)

Fault Tree Analysis can be said to be a general tool for reliability analysis. It can be

applicable not only simple-structured systems but also complex-structured systems. As the system under study gets bigger, the fault tree of the system gets bigger, but it does not grow exponentially. So, with help of some computer-aided tools, the analysis of a large and complex system can be performed in a reasonable time. And after generating cutsets in the fault tree of a system, critical components of the system are easily identified. But, with fault tree analysis it is hard to calculate the analytic solutions for the reliability of a system, so only probabilistic analysis like unavailability calculation is usually performed.

Monte Carlo Simulation

One of the most important strengths of Monte Carlo simulation is that it is not restricted by the structure of a system and much knowledge is not necessarily required for modeling and estimating the reliability of the system. But, in case the reliability of a system is very high, poor results can be obtained in a reasonable run time.

Considering possible inaccuracies of the basic failure rates of hardware and software components, and resulting reliabilities of subsystems, we are skeptical to try to obtain mathematically perfect analytic solution for the reliability of a system as a function of time. And also considering the complexity and high reliability of a system, we adopt fault tree analysis for the high-level phase of reliability calculation

IV. An Application – The Reliability of Dynamic Safety System

Until now, we separate the process of estimating the reliability of a digital system into two phases and adopt a proper approach for each of the two phases of the process. In this section, we present an application of our model to Dynamic Safety System(DSS) to demonstrate the feasibility of our model.

Description of Dynamic Safety System (DSS)

Fig. 2 shows the schematic diagram of a typical Dynamic Safety System (DSS)[7]. As shown in Fig. 2, DSS consists of 4 channels, each of which are independent to each other. In Fig. 2, MUX represents multiplexer, ADC means analog-to-digital converter, TAC means trip

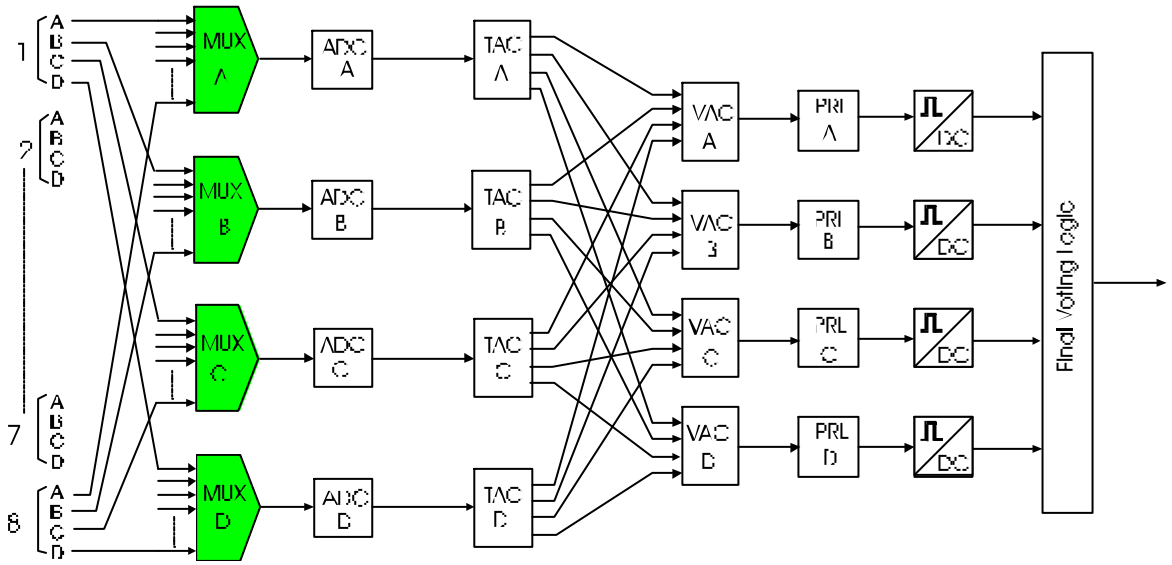

Fig. 2 Schematic diagram of a typical Dynamic Safety System [7]

algorithm computer, VAC means voting algorithm computer, and PRL means pattern recognition logic. Field measurement signals are transmitted to MUXs. MUXs sample the input signals and send the sampled signals to ADCs where analog-to-digital conversion is performed. TACs receive the digital values converted by ADCs and determine whether they should generate trip signals or not. The results of four TACs are sent to four VACs where 2-out-of-4 logic is performed for all trip states. The results of VACs are sent to PRLs where comparison between an expected pattern of trip states and the received pattern of trip states is performed. If the two patterns does not match to each other in some PRLs, the PRLs send trip signal to final voting logic which is selective 2-out-of-4 logic of trip circuit breakers(TCBs).

The originality of DSS design is its continuous active testing feature. By interleaving test signals which can cause TACs to generate trip signals into the normal operation signals and comparing the expected pattern of trip states to the received pattern of trip states, DSS checks its own operability continuously.

Low-Level Phase of Reliability Estimation

Considering the complexities of the subsystems of DSS, the dominant components for the reliability estimation are TACs, VACs and PRLs. And also considering the fact that transient faults are expected to be more often than permanent faults like the workstation example stated above, it is assumed that transient-fault-induced-failures are dominant in DSS. Therefore, it can be said that in estimating the reliability of DSS, it is enough to consider only transient faults and software faults.

We assume that TACs, VACs and PRLs have same hardware structures and same software faults. If we assume that the failure rate of each component in TACs, VACs, and PRLs is 10^{-6} /hr and the failure rate of each software in the components is also 10^{-6} /hr, the estimated failure rates of TACs, VACs and PRLs are commonly 1.144×10^{-6}/hr.[5]

High-Level Phase of Reliability Estimation

We adopt fault tree analysis for high-level phase of reliability estimation. Because our intention is to perform unavailability analysis with known transient-fault-induced-failure rates of subsystems, we first need to calculate the probabilities that subsystems are unavailable due to transient faults. The unavailability due to transient faults are calculated as follows:

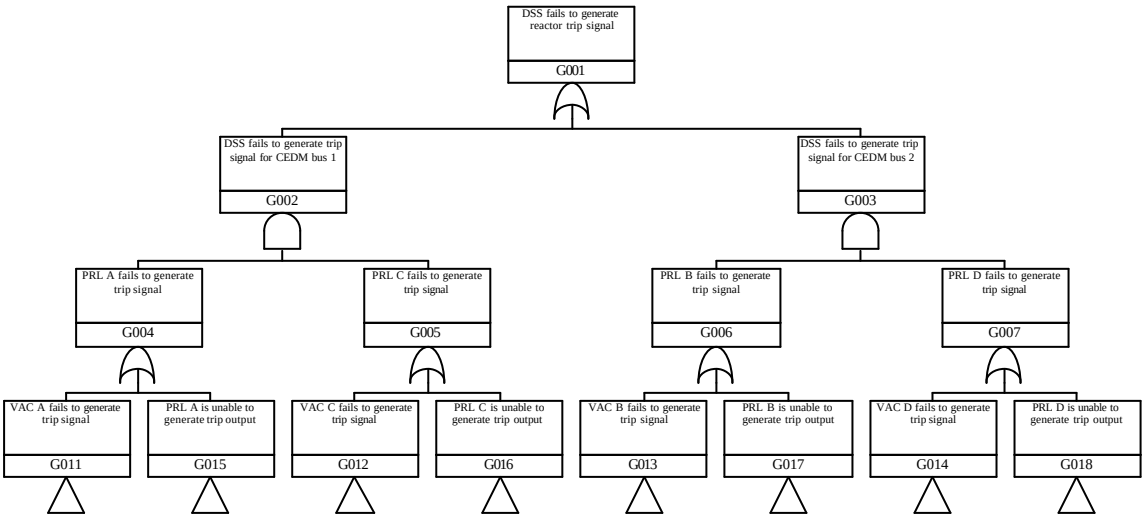

Fig. 3 Fault Tree for Dynamic Safety System

$$Q = \frac{l^t_{sys} T}{2} \quad (6)$$

where

Q : the unavailability of a subsystem

l^t_{sys} : the transient-fault-induced failure rate

T : testing interval (assume 25ms)

According to Eq.(6), the unavailabilities of TACs, VACs and PRLs are determined to be 3.96×10^{-12}. To take common cause failures(CCFs) into account, we assumed that there is additional 5% of probability that the components of same kind fail simultaneously. Fig. 3 shows the fault tree for Dynamic Safety System. Analysis results show that the overall system unavailability due to transient faults is 5.94×10^{-13}. Table 1 shows dominant cutsets in DSS which can be used to identify critical components of the system.

Discussions

In calculating the unavailability of DSS shown above, a few things which are important in the real world are missing, such as fail-safe design features, manual trip by human operator and repairing when a failure occurs. Even though we take all failure scenarios into consideration when calculating the unavailability of DSS, actually due to the fail-safe design features of DSS, not all of the failure scenarios need to be considered. We only need to consider the scenarios that DSS cannot generate trip signals when needed.

If we further consider manual trip by human operators, the unavailability of DSS would be further decreasing. If manual trip by operators are taken into account and human operators will

No.	Inputs	Description	Event Prob.	Cutset Prob.
1	FIOXPRLS	Common cause failure of PRLs	1.98e-13	1.98e-13
2	FIOXVACS	Common cause failure of VACs	1.98e-13	1.98e-13
3	FIOXTACS	Common cause failure of TACs	1.98e-13	1.98e-13
4	FICPPRLA	PRL A fails to provide output	3.96e-12	1.57e-23
	FICPPRLC	PRL C fails to provide output	3.96e-12	
5	FICPPRLB	PRL B fails to provide output	3.96e-12	1.57e-23
	FICPPRLD	PRL D fails to provide output	3.96e-12	
6	FICPPRLA	PRL A fails to provide output	3.96e-12	1.57e-23
	FIOVACDOC	VAC C fails to generate trip output	3.96e-12	
7	FICPPRLC	PRL A fails to provide output	3.96e-12	1.57e-23
	FIOVACDOA	VAC A fails to generate trip output	3.96e-12	
8	FICPPRLB	PRL A fails to provide output	3.96e-12	1.57e-23
	FIOVACDOD	VAC D fails to generate trip output	3.96e-12	
9	FICPPRLD	PRL A fails to provide output	3.96e-12	1.57e-23
	FIOVACDOB	VAC B fails to generate trip output	3.96e-12	
10	FICVACDOA	VAC A fails to generate trip output	3.96e-12	1.57e-23
	FICVACDOC	VAC C fails to generate trip output	3.96e-12	
11	FICVACDOB	VAC B fails to generate trip output	3.96e-12	1.57e-23
	FICVACDOD	VAC D fails to generate trip output	3.96e-12	

Table 1 Dominant Cutsets in Dynamic Safety System

succeed in manually tripping the reactor with the probability 95% when needed, the overall unavailability of DSS would decrease to 3.0×10^{-14}.

Repairing of failed components is not considered in the model described above. In the model above, when a component fails, it remains in the failed state. Therefore, the second failure of the system may lead DSS to our-of-service state. (for example, if PRL C fails after the failure of PRL A, DSS is unable to perform its protection function.) If failures are recognized by operators and failed components are repaired properly in reasonable time, the unavailability of DSS becomes much smaller.

Online active testing feature of DSS is also not included in the model described above. DSS continuously check its operability through interleaving test signals into its normal input signals. Mismatch of expected pattern of trip states and received trip states directly means the trip of that channel, so if two or more channels are not operable simultaneously, DSS automatically trips the reactor. If this active testing is perfect, DSS would trip the reactor and protect the power plant before the failures of its components cause any threat to the safety of the power plant.

It is worth to note that the unavailability results according to the model described above are still quite conservative. Even though recognizing the dominance of transient faults in digital systems is a major step to relieve severe conservatism in estimating the reliability of a digital system, it seems that there still remains a long way to go to obtain much more realistic estimation of the reliability of a digital system.

V. Conclusions

We developed an integrated model for estimating the reliability of a digital system. To take advantages of historical models developed so far, we separated the process of estimating the reliability into two phases, low-level phase and high-level phase. The boundary of two phases is the reliabilities of subsystems. In the low-level phase, the reliabilities of subsystems are estimated based on the structure of hardware and software components in subsystems and the interaction between the hardware and software components. In the high-level phase, the reliability of the system is calculated based on the reliabilities of subsystems estimated in the low-level phase. Fault tree analysis was adopted for the high-level phase of reliability estimation process.

As an example, the model is applied to the estimation of the reliability of Dynamic Safety System. Assuming that TACs, VACs, and PRLs have same structure and complexity, their failure rates were found to be 1.144×10^{-6}/hr. Based on this result, fault tree analysis is performed to calculate the unavailability of DSS. The unavailability of DSS is found to be 5.94×10^{-13}.

Considering the assumptions made in the estimation of the reliability of DSS, the result obtained according to the model tends to be still conservative. To get more realistic estimation of the reliability of a digital system, more things need to be taken into account in the model.

References

[1] Goel, A. L. and Soenjoto, J., Models for Hardware-Software System Operational Performance Evaluation, *IEEE Trans. on Reliability*, **R-30**(1981) 232-239

[2] Sumita, U and Masuda, Y, Analysis of Software Availability/Reliability Under the Influence of Hardware Failures, *IEEE Trans. on Software Engineering*, **SE-12**(1986) 32-41

[3] Welke, S. R., Johnson, B. W. and Aylor, J. H., Reliability Modeling of Hardware/Software Systems, *IEEE Trans. on Reliability*, **44**(1995) 413-418

[4] Vemuri, K. K. and Dugan, J. B., Reliability Analysis of Complex Hardware-Software Systems, *1999 Proceedings Annual Reliability and Maintainability Symposium*, (1999) 178-182

[5] Choi, J. G. and Seong, P. H., An Integrated Approach for Reliability Estimation of Nuclear Power Plant Digital Protection Systems: An application to Dynamic Safety System, *International Seminar on Software Reliability of Man-Machine Systems*, (2000) 19
[6] Siewiorek, D. P. and Swarz R. S., Reliable Computer Systems – Design and Evaluation 3d edition , A K Peters, Ltd., 1998
[7] Kim, U. S. and Seong, P. H., An Application of Dynamic Safety System to Pressurized Water Reactor, *Ann. Nucl. Energy*, **25**(1998) 1221-1233