Advantages of a Dynamic RGGG Method in Qualitative and Quantitative Analysis

Seung Ki Shin [a], Poong Hyun Seong [b*]
[a,b] *Department of Nuclear and Quantum Engineering, KAIST*
373-1, Guseong-dong, Yuseong-gu, Daejeon, Republic of Korea, 305-701
[a] *kankani@kaist.ac.kr*
[*] *Corresponding author: phseong@kaist.ac.kr*

1. Introduction

Various researches have been conducted in order to analyze dynamic interactions among components and process variables in nuclear power plants which cannot be handled by static reliability analysis methods such as conventional fault tree and event tree techniques. A dynamic reliability graph with general gates (RGGG) method was proposed for an intuitive modeling of dynamic systems [1] and it enables one to easily analyze huge and complex systems.

In this paper, advantages of the dynamic RGGG method are assessed through two stages: system modeling and quantitative analysis. And then a software tool for dynamic RGGG method is introduced and an application to a real dynamic system is accompanied.

2. System modeling

Modeling of a target system is performed for a qualitative analysis prior to a quantitative analysis. Methods for efficiently representing the salient characteristics of dynamic systems have been recognized as an important ingredient of dynamic reliability analysis [2].

The main advantage of the dynamic RGGG is to pave the way for an intuitive modeling method that can capture the dynamic behavior of a system failure. A one-to-one match from the actual structure of a system to dynamic RGGG can be achieved by conventional static nodes of the RGGG [3] and novel dynamic nodes introduced in the dynamic RGGG method. Figure 1 shows the dynamic nodes.

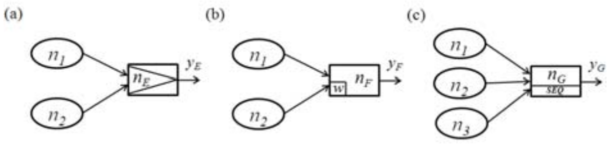

Fig. 1. The dynamic nodes of a dynamic RGGG: (a) a PAND; (b) a WSP; and (c) an SEQ node.

3. Quantitative analysis

For the quantitative analysis of the dynamic RGGG, it is converted into an equivalent Bayesian network by determining the probability tables of all the nodes and there is no need for a transformation of the model structure in the qualitative analysis stage.

A discrete-time method is used to evaluate the dynamic RGGG and detailed algorithms for making each probability table can be found in [1].

As the dynamic RGGG use discrete-time method it can avoid a state space explosion problem of Markov chain method which is widely employed for the dynamic reliability analysis. Let's assume that a dynamic system is composed of k components and the state of each component can be either success or failure. If we analyze this system using Markov method, 2^k states are needed. Therefore as k increases, the complexity of calculation increases exponentially. Whereas in the dynamic RGGG method when n is the number of time discretization, $(n+1)^3$ blanks should be filled in probability table of each node (component). Therefore the total of blanks in the RGGG is $(n+1)^3 \cdot k$ and as k increases, the complexity of calculation increases linearly with k, not exponentially. Consequently it has a great advantage especially when the target system is very complex.

4. A software tool for the dynamic RGGG

The accuracy of the dynamic RGGG method is limited due to the assumption of discrete time, but it can be ensured as the number of time discretizations increases. A software tool for the evaluation of the dynamic RGGG is being developed using the proposed algorithms, so the almost accurate results can be computed. The software tool is utilized to calculate a reliability of an example system in the following section.

5. Application to a cardiac assist system

To check the ability of the dynamic RGGG method, we give an example in this section. Figure 2 shows a block diagram of a cardiac assist system which was analyzed in [4, 5].

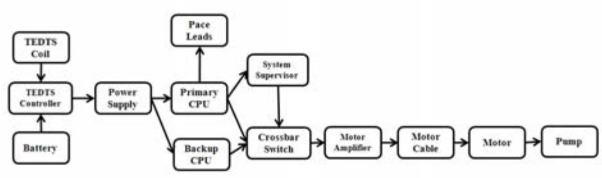

Fig. 2. Block diagram of a cardiac assist system.

This system has two dynamic properties; the backup CPU is a warm spare for the primary CPU, and failure of either the crossbar switch or the system supervisor fails both the primary and backup CPU. Figure 3 and 4 shows DFT and a dynamic RGGG constructed using the software tool for the example system respectively. From these two figures, we can check the intuitive modeling power of the RGGG method as it has almost same structure with the block diagram of the actual system in Figure 2.

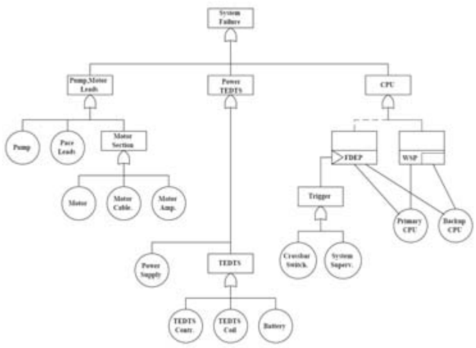

Fig. 3. Dynamic fault tree of a cardiac assist system.

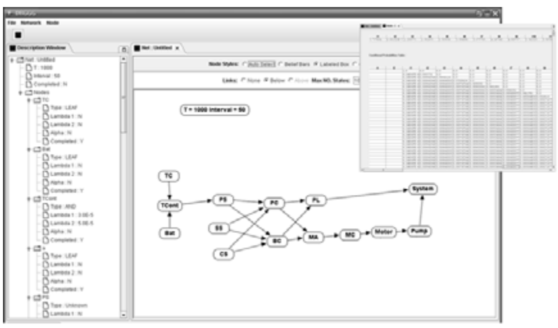

Fig. 4. Dynamic RGGG of a cardiac assist system.

The reliability of the system is computed as 0.6095442 by Markov chain method and results from the dynamic RGGG method using the developed software tool are shown in Table I. We can check that the error becomes smaller as the discretization number (*n)* increases; when *n* is *200*, the numerical difference is smaller than 10^{-5}.

Table I: Evaluation results of the dynamic RGGG

n	Reliability
10	*0.6096437*
20	*0.6095939*
50	*0.6095641*
100	*0.6095541*
150	*0.6095508*
200	*0.6095492*

6. Summary & conclusions

The static RGGG method was upgraded into the dynamic RGGG so that it can model dynamic systems intuitively. In this paper advantages of the dynamic RGGG method were discussed in two stages: system modeling and quantitative analysis, and then a software tool for the dynamic RGGG was briefly introduced. As the structure of RGGG is almost same as the block diagram of an actual system and it does not undergo the state space explosion problem, the dynamic RGGG method may have many advantages in qualitative and quantitative analyses of complex dynamic systems.

REFERENCES

[1] S. K. Shin and P. H. Seong, Review of Various Dynamic Modeling Methods and Development of an Intuitive Modeling Method for Dynamic Systems, Vol. 40, No. 5, pp. 375-386, 2008.

[2] P. E. Labeau, C. Smidts, and S. Swaminathan, Dynamic Reliability: Towards an Integrated Platform for Probabilistic Risk Assessment", Reliability Engineering and System Safety, Vol. 68, No. 3, pp. 219-254, 2000.

[3] M. C. Kim and P. H. Seong, Reliability Graph with General Gates: an Intuitive and Practical Method for System Reliability Analysis, Reliability Engineering and System Safety, Vol. 78, No. 3, pp. 239-246, 2002.

[4] Y. Ren and J. B. Dugan, Design of Reliable Systems using Static & Dynamic Fault Tree, IEEE Transactions on Reliability, Vol. 47, No. 3, pp. 234-244, 1998.

[5] Y. Ou, J. B. Dugan, Sensitivity Analysis of Modular Dynamic Fault Trees, Proceedings of IEEE International Computer Performance and Dependability Symposium, pp. 35-43, 2000.