INL/CON-23-71652-Revision-0

Analyzing Hardware and Software Common Cause Failures in Digital Instrumentation and Control Systems using Dual Error Propagation Method

July 2023

Priyanka Pandit, Arjun Earthperson, Mihai Diaconeasa, Han Bao

Changing the World's Energy Future

Idaho National Laboratory

INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, LLC

INL/CON-23-71652-Revision-0

Analyzing Hardware and Software Common Cause Failures in Digital Instrumentation and Control Systems using Dual Error Propagation Method

Priyanka Pandit, Arjun Earthperson, Mihai Diaconeasa, Han Bao

July 2023

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Analyzing Hardware and Software Common Cause Failures in Digital Instrumentation and Control Systems Using Dual Error Propagation Method

Priyanka Pandit[1], Arjun Earthperson[1], Han Bao[2], Mihai Diaconeasa[1,*]

[1]Department of Nuclear Engineering, NCSU, Raleigh, NC
[2]Idaho National Lab, Idaho Falls, ID

[leave space for DOI, which will be inserted by ANS]

ABSTRACT

This paper develops a methodology for quantifying software common cause failure parameters (CCFs) in nuclear power plants' digital instrumentation and control systems. Probabilistic Risk Assessment (PRA) techniques are used to support the transition of analog instrumentation and control systems to digital in nuclear power plants. The hardware components of the I&C systems have reliability databases that can be used in probabilistic risk assessment studies. However, the failure data for redundant software components of the systems is sparse. Failure of components constitutes a CCF, wherein two or more components or systems fail due to a single shared cause and coupling mechanism.

This paper proposes a quantification approach that can simultaneously model hardware and software components, incorporate the CCFs of software systems in the models, and bridge the gap between the failure quantification of models and the development of CCF parametric databases. We demonstrate the dual error propagation method by developing instrumentation and control systems failure models for a representative digital reactor trip system. The dual error propagation method models are built to simulate the control and data flows within the systems and can accommodate failure states. By expanding the dual error propagation method to software CCFs, we generated alpha factor parameter estimates for each modeled error propagation mechanism.

Keywords: software failures, common cause failures, probabilistic risk assessment, dual error propagation method.

1. INTRODUCTION

Instrumentation and control (I&C) systems are key in nuclear power plants. They monitor and control power plant parameters and prevent and mitigate accident conditions [1]. Analog I&C systems, currently used in nuclear power plants, will gradually become obsolete, replaced by digital I&C systems [2]. The development of risk assessment frameworks is underway to support the transition to digital I&C systems [3] [4] [5], but one main challenge is quantifying common cause failure parameters and probabilities (CCFs) in the digital I&C systems [6]. A CCF occurs when two or more components fail due to a single shared cause and coupling mechanism [7]. The digital I&C systems have redundant trains of identical components. In the case of a component failure, the redundant component can continue the system's function. However, the very nature of the redundant components makes them susceptible to CCFs.

NUREG-5485 gives guidelines to aid probabilistic risk assessment (PRA) analysts in modeling CCF using parametric models [7]. While the failure databases used in parametric models contain hardware failure information, they do not cover software failures [8]. In literature, numerous studies present methods to

* madiacon@ncsu.edu

quantify software failures in digital I&C systems. Shorthill et al. have adapted the systems-theoretic process analysis to explicitly incorporate redundancies for the hazard analysis of digital I&C systems in nuclear power plants [9]. Building on the hazard analysis results in [9], Shorthill et al. have developed the Bayesian and HRA (human reliability analysis)-aided method for the reliability analysis of software (BAHAMAS) for the quantification of software hazards[10]. Bao et al. have developed software CCF quantification methods for a limited-data and data-rich scenario[11]. Mohaghegh et al. and Sakurahara et al. use model-based simulation to quantify physical CCFs [12] [13]. Robert Brill examines the licensee event report database to investigate digital I&C failures [14]. Chu et al. review software-induced failures and present their insights into modeling them in PRA [15]. Bin Li and Dongfeng Zhu have made finite-state machines representing software elements as machine states [16] [17]. Finite state machines are a specific case of discrete Markov chains used in this paper's quantification approach. Andrey Morozov's thesis, "Dual Error Propagation Model for Error Propagation Analysis of Mechatronic Systems," presents a model-based simulation method for quantifying software CCF [18].

Even though there is extensive literature on software failures, CCF, model-based simulation to quantify CCF and model-based simulation for mechatronic systems, there is a need for a quantification approach that can achieve all the following:

- Develop models of software components and hardware components to quantify failure probability
- Incorporate the CCFs of software systems in the models
- Bridge the gap between CCF quantification in models and the development of CCF parametric databases.

In this paper, we apply the dual error propagation method (DEPM) for quantifying software CCFs by developing I&C systems failure models for a representative digital reactor trip system. The DEPM models are built to simulate the control and data flows within the systems and can accommodate failure states. By expanding DEPM to software CCFs, we generated alpha factor parameter estimates for each modeled error propagation mechanism. Section 2 of this paper describes CCFs and the alpha factor model for quantifying CCFs, along with presenting the DEPM. In Section 3, we present the quantification approach developed in this paper. Sections 4 and 5 present the conclusions and future work, respectively.

2. METHODOLOGY

This section presents the concepts and definitions used in the software CCF quantification approach presented in the paper.

1.1 Common Cause Failures

CCFs are defined as component failures that meet four criteria: (1) two or more individual components fail or are degraded, including failures during demand, in-service testing, or deficiencies that would have failed if a demand signal had been received, (2) components fail within a selected period such that success of the PRA mission would be uncertain, (3) component failures result from a single shared cause and coupling mechanism, and (4) a component failure occurs within the established component boundary [7]. NUREG-5485 presents the following parametric models to quantify CCFs:

- Alpha factor model
- Beta factor model
- Multiple Greek letter model
- Basic parameter model.

The beta factor model considers all the components to have failed; it is a single-parameter model. The multiple greek letter model is an extension of the beta factor model that includes more parameters. NUREG-5485 states that approximate parameter estimators have been developed for the beta factor and the multiple greek letter model, as it is difficult to obtain rigorous estimators. The alpha factor model is a multi-parameter model whose parameters are estimated from observable data from a sampling scheme. Hence the alpha factor model is recommended in NUREG-5485. In this paper, we use the alpha factor model to demonstrate the generation of model parameters.

1.1.1 Alpha Factor Model

The alpha factor model defines CCF probabilities from a set of failure frequency ratios and the total component frequency failure, Q_T. In terms of the basic event probabilities, the probabilities of a common cause basic event (CCBE) involving k specific components in a common cause component group of size m are given as,

for a staggered testing scheme,

$$Q_k^m = \frac{1}{\binom{m-1}{k-1}} \alpha_k Q_T \tag{1}$$

for a non-staggered testing scheme,

$$Q_k^m = \frac{k}{\binom{m-1}{k-1}} \frac{\alpha_k}{\alpha_T} Q_T. \tag{2}$$

The estimator for the alpha factor model involving k specific components in a common cause component group of size m is given as

$$\alpha_k = \frac{n_k}{\sum_{k=1}^{m} n_k} \tag{3}$$

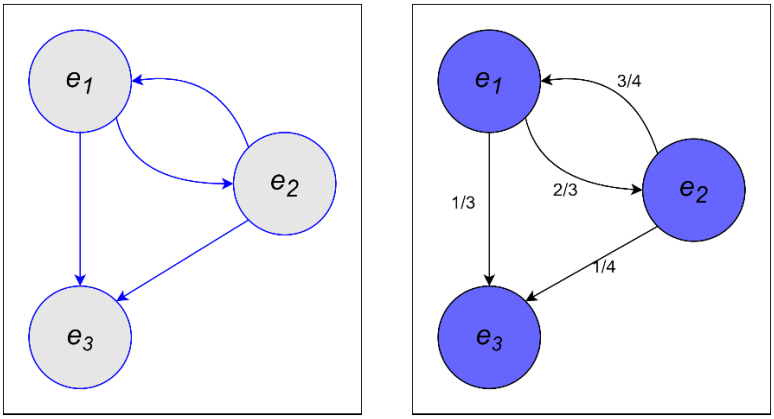
where n_k is the number of failures of a specific component, k.

1.2 Dual Error Propagation Method

The DEPM's purpose is to provide the possibility of simultaneous probabilistic analysis of control and data flow in the system under consideration [18]. In DEPM, two directed graph models are defined using the set of elements of a system: a data flow graph (DFG) and a control flow graph (CFG), as shown in
Figure 1. The nodes of both graphs represent the system elements. The DFG's arcs define the data transfer paths between the elements, which are also seen as the error propagation paths. The arcs of the CFG represent the control flow transitions between the elements, determining the order of their execution. The arcs of the CFG are weighted and show the probabilities of control transitions, like in a state graph of a discrete-time Markov chain. Faults are activated in the elements during their execution and result in errors. The occurred errors propagate to other elements through the data transfer and control flow paths. The error propagation between the elements is determined by the DFG and CFG structure. For example, the system shown in Figure 2 has a fault activated at the element e_1 with a probability of $EP_{e1} = 0.1$. Then the failure probability of the element e_7 is given by,

$$P_f(e_7) = P_{e1 \to e2} \cdot P_{e2 \to e5} \cdot P_{e5 \to e7} \cdot EP_{e1} = \frac{1}{2} \cdot 1 \cdot \frac{1}{2} \cdot 0.1 = 0.025. \tag{4}$$

Similarly, for state-based quantification, all the possible states of the system are enumerated. The failure probability is calculated by multiplying the probabilities of the arcs and, in case of multiple failure states, adding the products of the arcs, demonstrated in Section 3.



(a) Data Flow Graph (b) Control Flow Graph

Figure 1. DEPM Graph Models

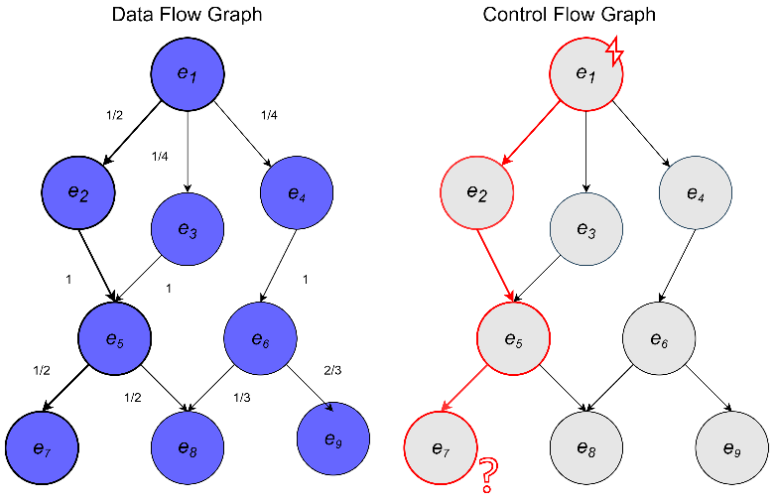


Figure 2. Quantification of DEPM models.

3. QUANTIFICATION APPROACH FOR SOFTWARE COMMON CAUSE FAILURES

We can demonstrate the proposed quantification approach using the example of two bistable processors (BPs). For every division of the reactor trip system, there are two redundant BPs. The function of a BP is to compare an incoming process variable from a sensor to a predefined setpoint and send a trip signal to the local coincidence logic processor if the process variable does not fit within the specified setpoint.
Figure 3 shows the internal functions of the BP.

The process variable is first converted from analog to digital signals. The bistable comparator algorithm then receives the digital process variable. For some process variables, the incoming signal should be less than the setpoint; for some process variables, the incoming variables should be greater than the setpoint.

Depending on the setpoint logic, the bistable comparator algorithm compares the process variables with their respective setpoints. It outputs a binary digital output, termed "low" and "high," for satisfies the condition and does not satisfy the condition. The trip algorithm's function is to send the trip signal to the local coincidence logic processors if the binary digital output from the bistable comparator algorithm is high.

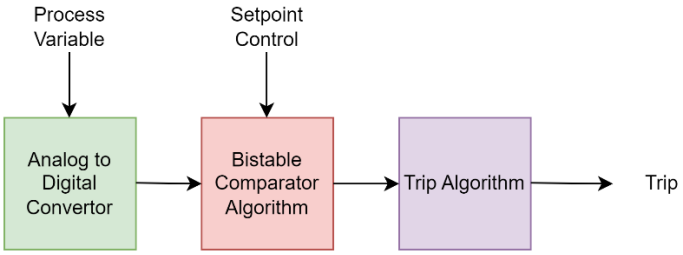


Figure 3. System diagram of the bistable processor.

Figure 4 shows the DEPM model for two BPs in one division of the APR1400 reactor trip system. To reduce the complexity in modeling, we have not considered the analog-to-digital converter; we instead assume that the process variable is in digital format when it comes to the bistable comparator algorithm. Table I gives the representations for the symbols used in Figure 4. The purple blocks represent elements, the black arrows represent control flow, the gray blocks represent data, the blue arrows represent data flow, and the red block represents a failure state. Table II sets up the DEPM model probability data, execution logic, and failure state conditions. The process variables used in the case study are hypothetical and are selected to simplify the demonstration.

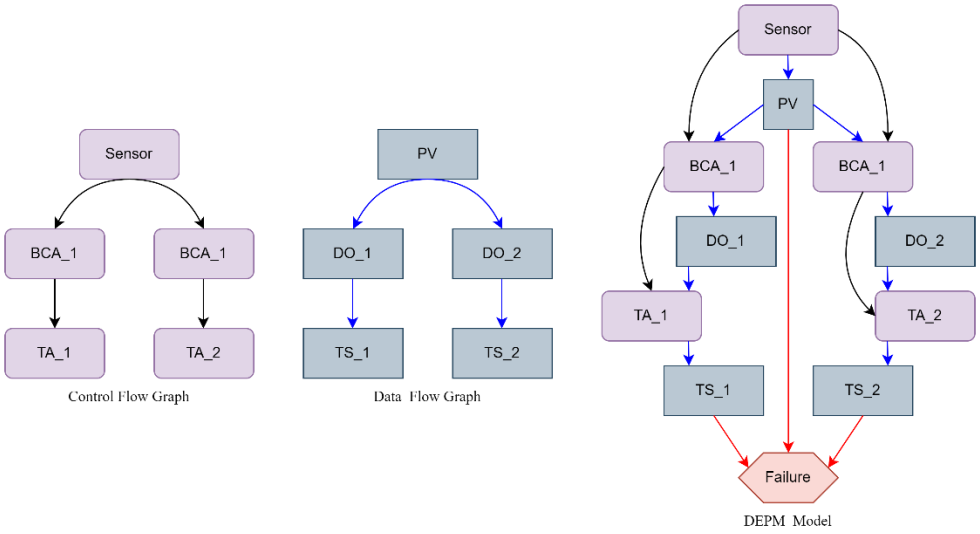


Figure 4. DEPM model for two bistable processors.

Table I. DEPM model acronyms.

Symbol	Representation
PV	Process Variable
BCA_1	Bistable Comparator Algorithm of BP 1
DO_1	Digital Output from BCA_1
TA_1	Trip Algorithm of BP 1

TS_1	Trip Signal from TA_1
BCA_2	Bistable Comparator Algorithm of BP 2
DO_2	Digital Output from BCA_2
TA_2	Trip Algorithm of BP 2
TS_2	Trip Signal from TA_2

Table II. DEPM model data.

Element/Data/Failure	Probabilities and Conditions.
Sensor	Control flow is initiated at the sensor and goes to elements BCA_1 and BCA_2.
PV	In this example, we consider the range of 10,11, and 12 units to be the process variables with probabilities of 0.333 each.
BCA_1	The setpoint is set at 12 units. If the PV is 10 or 11units, then DO_1 is low; if it is 12 units, then DO_1 is high.
DO_1	DO_1 has two states, high and low.
TA_1	If DO_1 is low, then TS_1 will be off. If DO_1 is high, then TS_1 is on.
TS_1	TS_1 has two states, on and off.
BCA_2	The setpoint is set at 11 units. If the PV is 10 units, then DO_2 is low, if it is 11 or 12 units, then DO_2 is high. It has an error probability of 0.1, which means that there is a 0.1 chance that the trip signal may be a false off or a false on.
DO_2	DO_2 has two states, high and low.
TA_2	If DO_2 is low, then TS_2 will be off. If DO_2 is high, then TS_2 is on.
TS_2	TS_2 has two states, on and off.
Failure	In this example, we set up the failure for the conditions of an incorrect setpoint being given to BCA_1. The correct setpoint is supposed to be 11 but is set to 12 in BCA_1. Hence, we can define failure as the condition that process variable is 11 units, TS_1 is off, and TS_2 is off. That is, we need one out of the two BCAs to function properly.

The state space for the DEPM model with two independent bistable comparator algorithms is shown in Figure 5. As shown in Figure 5, we end up with one failure state when the process variable is 11 units and the trip signal is off. From Table II, the failure probability of the failure state can be calculated as,

$$\mathrm{P_f}(11\text{units}, \mathrm{TS_1} = \text{off}, \mathrm{TS_2} = \text{off}) = \mathrm{P_{11\ units}} \cdot \mathrm{EP_{BCA_2}} = 0.333 \cdot 0.1 = 0.0333. \quad (5)$$

We can now analyze the same system with a common code for the bistable comparator algorithm. Since both the BPs have the same conditions coded in their respective comparator algorithms, we can use a single element BCA_1 to represent them, as shown in Figure 6 and Table III.

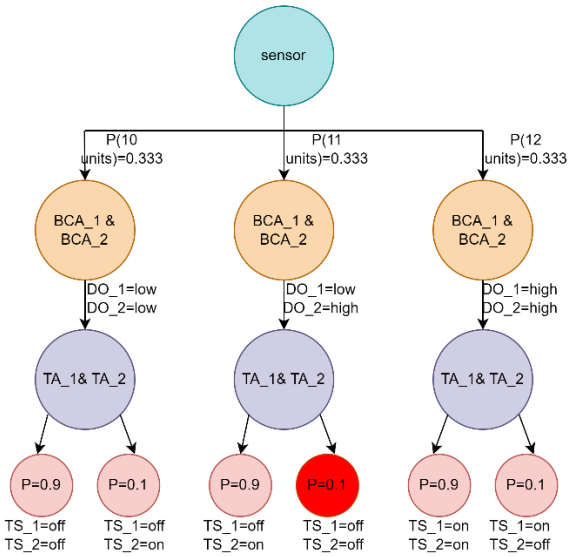


Figure 5. State space of the DEPM model of two bistable processors.

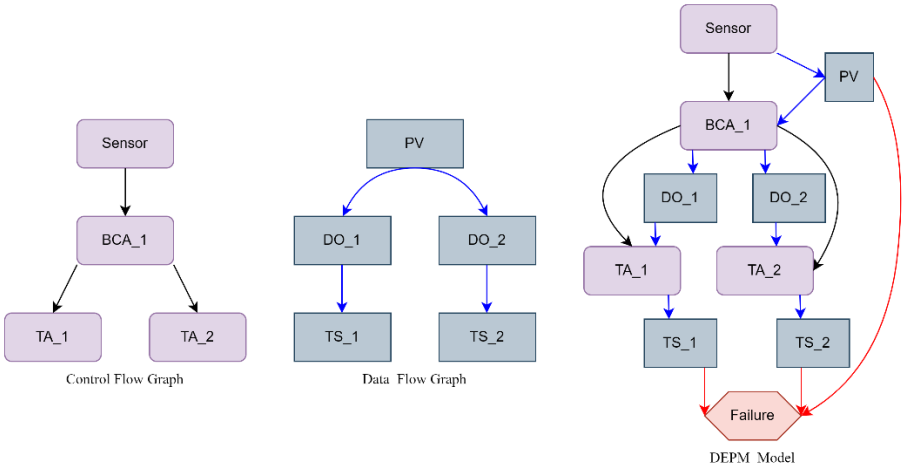


Figure 6. DEPM model for common bistable comparator algorithm among bistable processors.

Table III. DEPM model data for common bistable comparator algorithm among bistable processors.

Element/Data/Failure	Probabilities and Conditions.
Sensor	Control flow is initiated at the sensor and goes to elements BCA_1 and BCA_2.
PV	In this example, we consider the range of 10, 11, and 12 units to be the process variables with probabilities of 0.333 each.
BCA	The setpoint is set at 12 units. If the PV is 10 or 11units, then DO_1 is low; if it is 12 units, then the PV is high.
DO_1	DO_1 has two states, high and low.
TA_1	If DO_1 is low, then TS_1 will be off. If DO_1 is high, then TS_1 is on.
TS_1	TS_1 has two states, on and off.
DO_2	DO_2 has two states, high and low.
TA_2	If DO_2 is low, then TS_2 will be off. If DO_2 is high, then TS_2 is on.

TS_2	TS_2 has two states, on and off.
Failure	In this example, we set up the failure for the conditions of an incorrect setpoint being given to the BCA. The correct setpoint is supposed to be 11 but is set to 12. Hence, we can define failure as the condition that process variable is 11 units, TS_1 is off, and TS_2 is off.

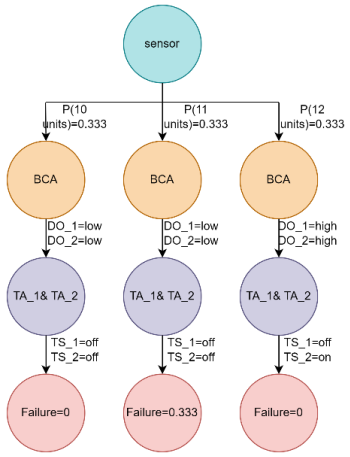


Figure 7. State space for DEPM model of bistable processors with a common bistable comparator algorithm.

We can see from Figure 7 that having a common software component between the BPs results in one branch of the state space satisfying the conditions for failure. The probability of the trip signal being off even when the setpoint is reached is 0.333, giving us the failure probability of one coupling mechanism in the BP. To calculate alpha factors ($\alpha_1 \ldots . \alpha_m$) for the CCF of an element (BPs in the case study) from the failure probabilities of the various coupling mechanisms (N) within the element that can cause CCFs, we use specific alpha factors ($\alpha_{1,1} \ldots . \alpha_{m,N}$), as given in the following equation. For N coupling mechanisms and a CCBE of size m,

$$\begin{matrix}\alpha_1 \\ \vdots \\ \alpha_m\end{matrix} = \begin{pmatrix}\alpha_{1,1} & \cdots & \alpha_{1,N} \\ \vdots & \ddots & \vdots \\ \alpha_{m,1} & \cdots & \alpha_{m,N}\end{pmatrix} / N. \tag{6}$$

Thus, we can use the alpha factor estimator for the coupling mechanism of an incorrect setpoint to get its specific alpha factors. We can consider the specific alpha factors as $\alpha_{1,1}$ and $\alpha_{2,1}$. Where $\alpha_{1,1}$ is the independent failure probability of the BPs without a common comparator algorithm and $\alpha_{2,1}$ is the specific alpha factor for the software CCF of the two BPs due to the incorrect setpoint in the common comparator algorithm. Using Equation 3, we get

$$\alpha_{1,1} = \frac{n_1}{\sum_{k=1}^{2} n_k} = \frac{0.0333}{0.333 + 0.1 \cdot 0.33} = 0.0909;\ \alpha_{2,1} = \frac{n_2}{\sum_{k=1}^{2} n_k} = \frac{0.333}{0.333 + 0.1 \cdot 0.33} = 0.909. \tag{7}$$

Similarly, we exhaustively analyze the probability of failures due to software coupling mechanisms in a system and get the specific alpha factors. Then for N coupling mechanisms in a CCBE of two BPs, we can obtain the alpha factors from equation (8). Some other potential coupling mechanisms for the bistable processors include incorrect sensor reading and trip algorithm error. Once we have the alpha factors from equation 8, we can calculate the component's total failure probability and common cause failure probabilities for staggered or non-staggered testing schemes, as given in equations (1) and (2).

$$\begin{matrix}\alpha_1 \\ \alpha_2\end{matrix} = \left(\begin{matrix}\alpha_{1,1} \\ \alpha_{2,1}\end{matrix} + \begin{matrix}\alpha_{1,2} \\ \alpha_{2,2}\end{matrix} + \cdots + \begin{matrix}\alpha_{1,N} \\ \alpha_{2,N}\end{matrix}\right)/N \quad (8)$$

4. CONCLUSIONS

This paper presented a methodology to obtain the parameters of the alpha factor model from system design, input parameters, and error propagation information. The model-based simulation methods we studied in the literature review focus either on representing the software as elements of a non-probabilistic model or using functions to quantify physical failure. In contrast, DEPM offers simultaneous probabilistic modeling of software and hardware components of the system. Compared to classical PRA methods that model only the failures of the I&C elements, DEPM can model the execution of the element's functions, where we can introduce and propagate errors to quantify the number of times we get a failure state. The DEPM lends itself to the modeling of CCFs due to the explicit representation of the software and hardware elements.

5. FUTURE WORK

For validation of results, the methodology presented in this paper needs to be applied to systems for which CCF parameters already exist. Further, the availability of operational and error propagation data required for DEPM modeling software CCFs must be investigated and consolidated. Regarding automating the methodology using code, OpenErroPro is the software package that implements the DEPM models [19].

ACKNOWLEDGMENTS

The authors thank Dr. Sai Zhang, Tate Shorthill, and Edward Chen for their guidance, discussions, and support during the writing of this paper. This work of authorship was prepared as an account of work sponsored by Idaho National Laboratory (under Contract DE-), an agency of the U.S. Government. Neither the U.S. Government, nor any agency thereof, nor any of their employees make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

REFERENCES

1. National Research Council. Digital Instrumentation and Control Systems in Nuclear Power Plants: Safety and Reliability Issues. Washington, DC: The National Academies Press (1997). https://doi.org/10.17226/5432.
2. IAEA, "Instrumentation and Control (I&C) Systems in Nuclear Power Plants: A Time of Transition," NTR2008 Supplement; https://www.iaea.org/sites/default/files/gc/gc52inf-3-att5_en.pdf (current as of Feb. 17, 2023).
3. "The benefits of digital I&C," Nuclear Engineering International; https://www.neimagazine.com/features/featurethe-benefits-of-digital-ic/ (current as of Feb. 15, 2023).
4. “Gösgen: From analog to digital instrumentation and control,” Framatome - Espace presse, 10 / 7:00 2019. https://www.framatome.com/medias/gosgen-from-analog-to-digital-instrumentation-and-control (current as of Feb. 15, 2023).
5. S. A. Arndt, "Digital Instrumentation and Control Systems Upgrades in Current Generation Nuclear Power Plants," 18th International Conference on Nuclear Engineering, Xi'an, China, Jan. 2010, Vol. 1, pp. 903–910, The American Society of Mechanical Engineers (2011). https://doi.org/10.1115/ICONE18-30358.

6. International Atomic Energy Agency (IAEA), Protecting against Common Cause Failures in Digital I&C Systems of Nuclear Power Plants, Nuclear Energy Series, No. NP-T-1.5, IAEA, Vienna, Austria (2009). https://www-pub.iaea.org/MTCD/Publications/PDF/Pub1410_web.pdf.
7. A. Mosleh, D. Rasmussen, and F. Marshall, "Guidelines on Modeling Common-Cause Failures in PRA," NUREG/CR-5485, Nuclear Regulatory Commission (Nov. 1998). https://nrcoe.inl.gov/publicdocs/CCF/NUREGCR-5485_Guidelines%20on%20Modeling%20Common-Cause%20Failures%20in%20PRA.pdf.
8. "Operating Experience Results and Databases," Nuclear Regulatory Commission; https://nrcoe.inl.gov/ccf_pe (current as of Feb. 17, 2023).
9. T. Shorthill, H. Bao, H. Zhang, and H. Ban, "A Redundancy-Guided Approach for the Hazard Analysis of Digital Instrumentation and Control Systems in Advanced Nuclear Power Plants," Nucl. Technol., **208**, 5, 1–20 (2021). https://doi.org/10.1080/00295450.2021.1957659.
10. T. Shorthill, H. Bao, H. Zhang, H. Ban. "A Novel Approach for Software Reliability Analysis of Digital Instrumentation and Control Systems in Nuclear Power Plants", Annals of Nuclear Energy, 158, pp. 108260, (2021).
11. H. Bao, H. Zhang, T. Shorthill, E. Chen, S. Lawrence. "Quantitative Evaluation of Common Cause Failures in High Safety-significant Safety-related Digital Instrumentation and Control Systems in Nuclear Power Plants", Reliability Engineering & System Safety (2022). https://doi.org/10.1016/j.ress.2022.108973.
12. Z. Mohaghegh, M. Modarres, and A. Christou, "Power2011-55324 Physics-Based Common Cause Failure Modeling in Probabilistic Risk Analysis: A Mechanistic Perspective," ASME 2011 Power Conference, Denver, Colorado, July 12–14, 2011, Vol. 2, pp. 201-210, The American Society of Mechanical Engineers (2011). https://doi.org/10.1115/POWER2011-55324.
13. T. Sakurahara, G. Schumock, S. Reihani, E. Kee, and Z. Mohaghegh, "Simulation-Informed Probabilistic Methodology for Common Cause Failure Analysis," Reliab. Eng. Syst. Saf., **185**, pp. 84–99 (2019) https://doi.org/10.1016/j.ress.2018.12.007.
14. R. W. Brill, "Instrumentation and Control System Failures in Nuclear Power Plants." Nuclear Regulatory Commission; https://www.nrc.gov/docs/ml0037/ML003757315.pdf (current as of Mar. 1, 2023).
15. T. L. Chu, G. Martinez-Guridi, M. Yue, and J. Lehner, "A Review of Software-Induced Failure Experience," BNL-NUREG-77124-2006-CP, Brookhaven National Laboratory (Nov. 2006) https://www.bnl.gov/isd/documents/32718.pdf.
16. B. Li, "Integrating Software into PRA," Thesis, University of Maryland (2004). https://drum.lib.umd.edu/bitstream/handle/1903/1993/umi-umd-1946.pdf?sequence=1&isAllowed=y.
17. D. Zhu, "Integrating Software Behavior Into Dynamic Probabilistic Risk Assessment," Thesis PhD, University of Maryland (2005).
18. A. Morozov, "Dual-graph model for error propagation analysis of mechatronic systems," PhD Thesis, Dresden University of Technology (2012).
19. M. S. Analysis, “mbsa-tud/OpenErrorPro.” Git Hub; https://github.com/mbsa-tud/OpenErrorPro (current as of Feb. 20, 2023).