A Review of Recent Dynamic Reliability Analysis Methods and a Proposal for a Smart Component Methodology

Darpan Krishnakumar Shukla and A. John Arul

Abstract The next-generation nuclear power plants are designed to have inherent safety features and passive safety systems and use advanced digital instrumentation and control (IC) systems to achieve required operational performance and to meet the safety goals. Digital IC systems often perform complex tasks while interacting with process dynamics. Static reliability methodologies have been shown to be inadequate for modeling and accurately estimating the reliability of such systems due to the lack of close correspondence between the model and the system. Though a number of new dynamic methods have been developed for the reliability evaluation of such systems, they have not reached a stage of maturity, like that of the traditional event tree/fault tree methods. Therefore, an effort is made to review the diverse and recent development in the field, with a view to identify suitable attributes and methods which are necessary for it to be widely adopted and for which a general-purpose software tool could be developed. The qualitative analysis is performed for the attributes among the reviewed methods for comparison and identifying the best method. A qualitative comparison is given for the major attributes. The paper recommends and outlines the development of smart component methodology (SCM) for reliability modeling of dynamic safety systems.

Keywords Dynamic reliability · Probabilistic dynamics · Reliability · Smart component methodology

D. K. Shukla (✉)
Indira Gandhi Centre for Atomic Research,
Homi Bhabha National Institute, Kalpakkam 603102, Tamil Nadu, India
e-mail: darpanks@igcar.gov.in

A. John Arul
Indira Gandhi Centre for Atomic Research, Kalpakkam 603102, Tamil Nadu, India
e-mail: arul@igcar.gov.in

P. V. Varde et al. (eds.), *Reliability, Safety and Hazard Assessment for Risk-Based Technologies*, Lecture Notes in Mechanical Engineering,
https://doi.org/10.1007/978-981-13-9008-1_22

1 Introduction

Real systems are complex interacting entities, where the performance requirements, system configuration and failure parameters may change with time. With the advent in digital instrumentation and control (IC) technologies, the next-generation nuclear power plants are expected to be built with inherent safety features, passive performance and self-diagnostics. Static reliability methods, though widely used, do not adequately model such systems exactly using its pure combinatorics. This gap between the reliability model and the system is claimed in the literature, will be reduced if we carry out *dynamic reliability* evaluation for the system, and will get the results of our interest. While doing dynamic reliability evaluation of the system, physical process, hardware, software, human performance, etc., can be considered precisely in the system model for failure analysis; a methodology is followed for system structuring and for quantifying it, which is called dynamic reliability method. In this section, we summarize some of the reported definitions of dynamic reliability and then mention some of the challenges to be checked for reviewing the methods. Some views on the definition of dynamic reliability and dynamic reliability methodologies found in the literature are presented below.

Devooght, J. [1]: Dynamic reliability accounts the dynamic nature of system, i.e., the evolution of the system, change in dynamics due to failures, repairs, maintenance, etc., and hence, change in failure rates due to new dynamics.

Labeau, P. E. et al. [2]: Dynamic reliability methods provide a framework for explicitly capturing the influence of time and process dynamics on scenarios.

Aldemir, T. [3]: Dynamic methodologies for PSA defined as a time-dependent phenomenological model of system evolution along with its stochastic behavior to account for possible dependencies between failure events.

Babykina, G. et al. [4]: In the context of safety analysis, dynamic reliability can be seen as an extension of system reliability assessment methods to the case in which the structure function changes in time with a discrete evolution (e.g., in the case of a phased mission system) and with a continuous evolution (e.g., in the case of dependency between the reliability of components and continuous physical variables). The structure function is meant to be a function describing the link between the states of system components, usually represented by their failures and repairs, and the state of the system itself. The dynamic reliability accounts for the following phenomena: (1) dynamic behavior of system components (aging and fatigue of different types), (2) importance of the order of occurring events (ordered sequences of events are considered in place of cut sets) and (3) multiple natures (stochastic and deterministic) of events driving the transitions between states.

To summarize the definitions, *dynamic reliability methods* must be able to model

1. Time dependence of system structure function.
2. Time ordering of basic events.
3. Time dependence of the reliability parameters (failure rate, mission time, test interval).

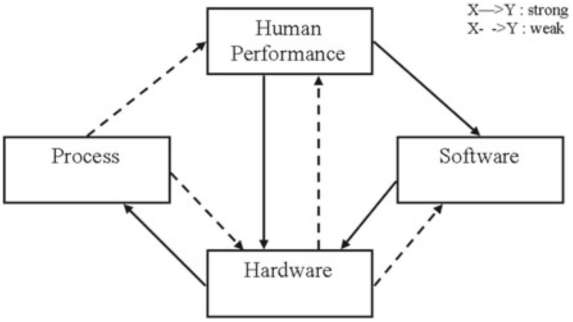

Fig. 1 Possible interaction between entities

4. Interaction effects: inter-dependence between process variables, hardware states, software function and human performance.

Hence, dynamic reliability methods give detailed modeling capability for the dynamic systems having interactions as described in Fig. 1, where the dotted and strong arrows indicate weak and strong interactions between the entities, respectively. In contrary to the benefits offered by the dynamic reliability methods of taking care of the interactions and other dynamic aspects in the method rather than the dependency on the analyst, which is for his careful approximation in static reliability analysis, the dynamic reliability analysis is challenged with the handling of complexity of the system, computational intensive nature and easiness of usage of the method. Various dynamic reliability methods have been developed in the last three decades and applied to many dynamic scenarios. In the literature, there is no consensus on the acceptance of a method for dynamic reliability assessment; the newly developed methods have not reached the maturity of application, like that of the traditional event tree/fault tree methods.

Therefore, the methods available in the literature are reviewed and compared with a view to identify suitability with respect to the attributes of accuracy, simplicity, scalability, fidelity and generality of modeling. However, we declare that the review does not aim to be exhaustive and complete but to find out a learned recommendation which would be a promising step toward a solution for all the above-mentioned problems. Based on the review, the paper recommends and outlines the peculiar qualities of smart component-based methodology along with Monte Carlo simulation for addressing the problems of dynamic reliability.

The paper is organized as follows. Section 2 presents the review of some of well-known dynamic reliability methods and approaches. Section 3 proposes a s-mart component methodology. Section 4 presents a comparative evaluation of the various methods presented in Sect. 2. The paper is concluded in Sect. 5.

2 Existing Methods, Approaches and Tools for Dynamic Reliability

A number of dynamic methodologies have been developed for application ranging from small- to large-scale systems. A large number of review papers have been published in the recent past on the subject of dynamic reliability and its methodologies [1–3, 5]. In the last decade, significant reports have been published by USNRC on the reliability analysis of digital I&C systems. The report Ref. [6] presents a study of a number of methods for reliability modeling of digital I&C systems and its incorporation into existing probabilistic risk assessment (PRA) models. The study identifies two types of dynamic interactions that need to be considered while modeling digital I&C system reliability: Type I—interactions between the reactor protection and control systems and the controlled plant process and Type II—interaction between the components of the reactor protection and control systems itself. According to the report, the required minimum criteria for acceptance of dynamic methodologies (only seven major requirements are reproduced out of eleven) are [6]:

1. The methodology should account for both Type I and Type II interactions.
2. The model must be able to predict encountered and future failures well and cannot be purely based on previous experience.
3. The model must make valid and plausible assumptions, and the consequences of violating these assumptions need to be identified.
4. The data used in the quantification process must be credible to a significant portion of the technical community.
5. The model must be able to differentiate between a state that fails one safety check and those that fail multiple ones.
6. The model must be able to differentiate between faults that cause function failures and intermittent failures.
7. The model must have the ability to provide uncertainties associated with the results.

From the qualitative analysis provided for some of the methods against the eleven criteria in Ref. [6], it is concluded that none of the methods meet all the criteria; however, each of the different methods may be better suited for specific problems. USNRC-6901 has ranked Markov/cell-to-cell mapping technique (CCMT) method and dynamic flowgraph methodology (DFM), as the top two methods, with most positive features and least negative or uncertain features (using subjective criteria based on reported experience) for dynamic probabilistic safety assessment. The report also mentions that there is no regulatory requirement for a single methodology to be applicable to all digital I&C systems. However, availability of such a methodology will be a convenience [6].

Subsequent report USNRC-6942 (2007) [7] and 6985 (2009) [8] proof checked the two identified methods. The study also applied the methods to benchmark systems and demonstrated the incorporation of the methods into PRA procedures. Report USNRC-6962 (2008) [9] presented the limitations of traditional reliability methods for digital system reliability evaluations. However, we observe a state explosion problem for Markov/CCMT method and doubt the solvability of dynamic flowgraph methodology when applied to industrial-scale systems.

The following section briefly reviews some of the significant methods with a view to assess its suitability for general applicability to dynamic reliability analysis, flexibility for deriving static reliability analysis and suitability for industrial-scale problems.

2.1 Methods Based on Chapman–Kolmogorov Equation

2.1.1 Chapman–Kolmogorov Equation

A number of methods are based on the general mathematical description of the time evolution of the joint probability of hardware state and process variables. The *Chapman–Kolmogorov equation (C-K equation)* for probability mass transport is described in several Refs. [1, 10–13], but first introduced in Refs. [14] as a dynamic approach for modeling failure in process control systems using Markovian assumption. Then, the duo of probabilistic and deterministic behavior of the reactor dynamics is mathematically supported by Ref. [11, 12] as a *theory of probabilistic dynamics (PD)/continuous event tree*. Briefly, the C-K equation is described; herein, integro-differential form and its more useful integral forms for reliability and availability problems for application to dynamic PSA are found in Refs. [11, 15].

Let, the reactor process variables be denoted by $X = [x_1, x_2, x_3, \ldots x_p]$, where p is the number of process variables under consideration which are functions of time. The system hardware (H/W) logic state is determined from the component states of the system. Let us denote the hardware states by i, $i = 1, 2, \ldots q$, then the evolution of the process variables of the system is given by the following differential equation:

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = f_i(X), \quad \text{with} \quad X(0) = X_0, \quad \text{and} \quad X \in R^p \tag{1}$$

This is a complex nonlinear functional equation, the solution of which is given by

$$X = g_i(t, X_0), X_0 = g_i(0, X_0) \tag{2}$$

In the equation, function $f_i(X)$ describes the dynamic evolution of the process variables for a given H/W state i, which may have differential dependence on X. Hence, the overall state of the reactor dynamics is represented by (X, i). The system evolves either due to the change in state of the components (stochastic) or due to automated control action (deterministic). Generally, the probability of transition from one state to another depends mainly on X. Hence, $P(i \to j|X)$ is the conditional probability of transition from H/W state i to the H/W state j per unit time. Let $\Pi(X, i, t)$ be the probability density that the system is in state (X, i) at time t. Then, the C-K equation can be written as follows:

$$\begin{aligned}\frac{\partial \Pi(X, i, t)}{\partial t} &= \sum_{j \neq i} P(j \to i|X)\Pi(X, j, t) \\ &\quad - \sum_{j \neq i} P(i \to j|X)\Pi(X, i, t) - \nabla_X . f_i(X)\Pi(X, i, t) \qquad (3)\end{aligned}$$

Or

$$\frac{\partial \Pi(X, i, t)}{\partial t} = \sum_{j \neq i} P(j \to i|X)\Pi(X, j, t) - \lambda_i \Pi(X, i, t) - \nabla_X . f_i(X)\Pi(X, i, t) \qquad (4)$$

where $\lambda_i(X) = \sum_{j \neq i} P(i \to j|X)$ is defined as the total probability of outgoing transition from state i. It is assumed here that the probability Π and function f_i are well behaved, so that the divergence operation exists. There are situations when considering nonlinear systems, where the divergence may not exist and such special conditions are addressed in [16]. The special issues are not addressed in this paper as the focus here is on the development of an overall framework.

The integral version of the above equation is as follows [15]:

$$\begin{aligned}\Pi(X, i, t) &= \int \Pi(u, i, 0)\delta(X - g_i(t, u))e^{-\int_0^t \lambda_i(g_i(s,u))\mathrm{d}s}\mathrm{d}u \\ &\quad + \sum_{j \neq i} \int P(j \to i|u)\mathrm{d}u \int_0^t \delta(X - g_i(t - \tau, u))e^{-\int_0^{(t-\tau)} \lambda_i(g_i(s,u))\mathrm{d}s}\Pi(u, j, \tau)\mathrm{d}\tau \qquad (5)\end{aligned}$$

The integral equation describes that the probability density that the system is in state (X, i) at time t is the sum of two probabilities [15]: The first term represents the evolution of system in the same state i for all the time t without any transition, while the process variables are changing deterministically according to the dynamics. The second term describes all the transition into state i taking place before time t.

The C-K equation takes as input, transition probabilities (failure rates, repair rates), a set of process variables and related dynamics, domain of safe working of the system and initial distribution of probability density. The dynamic methodologies can further be classified according to input representation and solution methods for C-K equation. Numerical treatment of the C-K equation invokes inevitable discretization schemes. To reduce the complexity in solving the complete C-K equation, the process variables are divided into discrete cells and the most famous method applying this technique is the cell-to-cell mapping technique method. Some of the well-known solution schemes to the C-K equation are Monte Carlo algorithms [12], cell-to-cell mapping technique [17] and dynamic event tree techniques, which are reviewed in forthcoming sections. Dynamic event trees and classical event trees are also derived from continuous event tree which are essentially discrete time versions of continuous event tree [10] and are compared with each other in Refs. [15, 18].

2.1.2 Cell-to-cell Mapping Technique (CCMT) Methods

The *cell-to-cell mapping technique (CCMT)* for reliability and safety studies was first developed by Aldemir et al. [14, 17, 19, 20]. In the CCMT, the continuous process variables, present in the C-K equation, are divided into cells of smaller intervals. The representative C-K equation for CCMT is obtained from the general C-K equation by integrating over each of the cells. The probability density is assumed to be uniform in these cells. The discrete hardware states and the discretized process variables form a discrete state space. The interval boundaries of cells are preferentially placed at the limit values of the control action which will subsequently be useful for considering non-temporal distribution. The resulting Markov chain coupled with dynamic evolution equations is solved with time step Δt assuming that system will not change its state during the time step. The method can be used both in inductive mode, to identify system evolution with given initial conditions, and in deductive mode, to identify sequence of paths that lead to the top event/undesired system states [19, 21–23].

The Markov/CCMT method is attributed to have some inherent limitations from memory and computational time requirements. For storing and handling of the transition matrix, large memory is required. The required memory space is proportional to the number of process variables times the number of cells (into which each variable is discretized) times the number of states. Reducing the number of cells partitioning the safety domain will reduce the matrix size, but consequently accuracy will be reduced. Therefore, attempts have been made to reduce memory requirements for this method in [17, 24] by using the vectorization method and sparse matrix technique. Hassan [19] has provided improvements to the method for storage requirements by using database for storing process evolution. The limitation of the number of acceptable process variables for tractable treatment of the dynamics remains unsolved till now [2]. Next is the treatment of the transition rates which are dependent on the values of the process variables and the choice of the time step. The time step in Markov/CCMT should be so small that not more than one cell crossing should occur in that step. But, keeping a small time step will, first, underestimate the transition probability due to the small probability of leaving the cell boundary and, second, slow down the computations [25]. Reference [25] describes the process-dependent time step limitation as a modeling problem of CCMT; i.e., the dynamics underlying the whole process is incorrectly represented if the size of the cells is not reduced in correspondence to the value of the time step. Reference [26] has addressed the time step problem of CCMT by treating time as a continuous variable in continuous time cell-to-cell mapping technique (CCMT). Reference [27] treats the stiffness of the transition matrix which arises due to the difference in scales of the time constants of the process dynamics. Recently, Ref. [23] addresses the limitations of computational time and storage using backtracking algorithm with pruning mechanism. The reference also mentions that accuracy can be maintained by more sampling in the cells, and computational time can be managed by going for parallel computation. The Markov/CCMT is applied for modeling of BWR/6 SBLOCA [19, 28] and a benchmark system of digital feedwater control system of a PWR [29, 30], and hence it is a strong candidate for large-scale application [23].

2.2 Monte Carlo Simulation

2.2.1 Direct Monte Carlo Simulation

The first paper on the *Monte Carlo method (MC method)* was published in 1949 by N. Metropolis and S. Ulam. The MC method was initially applied to neutron transport calculations, and then the usage of the term Monte Carlo became synonymous with the term simulation, i.e., Monte Carlo simulation (MCS), while its applications in particle physics, communication network traffic, models of conflicts, computation of multiple integral, etc., have been explored in a wide range of the literature. For use in probabilistic dynamics, three slightly different MCS approaches, viz. discrete event simulation [5], dynamic Monte Carlo availability model (DYMCAM) [31] and analog MC [12], are used. The analog MC is the first application of Markov MC [32]. The analog MCS algorithm described in Ref. [12] is as follows:

Step-1 The initial configuration i and a value of the process variables X are sampled from $\Pi(X, i, 0)$.

Step-2 The next transition time t out of i is sampled.

Step-3 The evolution of X in state i is computed up to t; if a border of 'working' domain is crossed, or if the end of the accident duration is reached, the current history is stopped.

Step-4 A new state j is sampled; if j is an unacceptable configuration, the history is stopped.

Step-5 This procedure is repeated from Step-2, j being the new value of i.

The well-known advantages of MC method over other methods for application to probabilistic dynamics include: The method is insensitive to the dimensionality of the problem, MCS can treat all types of distributions for modeling the system, and MCS directly estimates the safety and reliability characteristics.

In contrast to the advantages, (i) the accuracy of results from the MCS method depends on the number of simulations performed, which may lead to large computational cost when it applies to PD, because after each sampling of the sequence of events the simulation calls the evolution function of the system according to the new configuration; (ii) in case of very low probability of failure events, most of the histories will not result in failures; hence, the number of histories required for good results could be huge. Consequently, increasing the number of simulations increases the computational time due to repeated calculations of deterministic evolutions, and decreasing the number of simulations decreases accuracy; hence, strong solution schemes are required.

2.2.2 Methods for Improving MCS

To improve the computational efficiency of MCS, two aspects need to be addressed: (i) reducing the number of samples required for a given accuracy and (ii) reducing the simulation time for each sample. Several methods have been developed for increasing the efficiency of statistical sampling for rare event simulation and improving deterministic simulation efficiency for repeated calculations. In the reliability estimation domain for reduction of variance of the result, importance sampling, subset simulation, biased simulation, forced transition, memorization are some of the acceleration techniques used Refs. [33–37]. The first time use of extreme value theory for rare event probability estimate is presented recently in Ref. [38]. The generalized extreme value distribution can be used to estimate the probability of crossing the safety boundary. Since there are only three asymptotic forms for extreme value distributions, they can be determined from limited simulations and can be used to extrapolate the estimate of reliability and failure frequency.

For reducing the process simulation time per sample, *pre-simulation memorization*-based schemes introduced in Refs. [39–42] are used. *Cell-to-boundary (CTB)* and *most probable evolution (MPE)*-based methods provide significant acceleration to the simulation. In CTB-based memorization method, the safety domain, D, is divided into cells and the evolution of system from all nodes is performed for all system configurations up to the closest boundary of the safety domain. The characteristics of these system evolutions are memorized. The characteristics include time and the value of X at the intersection with the closest boundary, the type of event encountered (control action/failure) at the boundary and a measure of the probability of the system to stay in the current state i up to the boundary. During the simulation, the time to the next transition is sampled from the current point in the process variable space. From the memorized time to the closest boundary from the neighboring nodes, since probability of reaching to the boundary is high, the dynamic calculations utilizing the memorization are brought to the boundary through an interpolation procedure [43]. The drawback of the memorization-based method is that if the size of the problem is large then memory requirements increase and the memorization becomes difficult.

In MPE, one defines a most probable set of initiating events; then for each of them, the dynamics of the system is integrated and main characteristics are stored till the end of the event sequence. The memory required to store is less compared to CTB method as least probable evolution paths are not calculated in MPE method [40]. Moreover, the use of a combination of CTB and MPE seems to be a more advantageous scheme. In a way, MPE is used to know the most probable evolution path, and cell in the neighborhood of the MPE paths is evaluated for CTB characteristics. The advantage of both the methods is brought together in Ref. [39]. In addition to these techniques, there is also the possibility of using response surfaces and metamodels for reducing the process evolution computation time [44].

The development of software tools based on MC method for dynamic reliability is based on hybridization of the MCS technique with other techniques which solve the problems highlighted in the previous paragraph. For example, the combination of CTB and MPE is described in Ref. [39].

Piecewise deterministic Markov process (PDMP) was first introduced by Davis in 1984 for Markov processes consisting of deterministic processes and random jumps. PDMP can be used for treating multiple failures [45] and preventive maintenance. The advantage of PDMP is that it is applicable to most types of dynamic reliability problems. However, PDMP is not easy to manipulate. It is difficult both to specify and to solve by methods other than MCS. The general PDMP algorithm for MCS is presented in Ref. [46] for dependability analysis of the famous heated tank system benchmark [14].

2.3 Dynamic Event Tree (DET)

In comparison with conventional event tree, dynamic event tree synthesizes branching at different time points and physical process evolution. It is often called as *discrete dynamic event tree (DDET)* to distinguish DET and CET. The first ever dynamic method introduced in 1981 for probabilistic transient analysis for PRA is called: event sequence and consequence spectrum using logical analytical methodology in Ref. [47], the first version of dynamic logical analytical methodology (DYLAM) [48]. Subsequently, many variants of DET-type methodologies and software tools that use DETs have been developed in USA and Europe, such as DYLAM-3 [48], dynamic event tree analysis method (DETAM) [49], accident dynamic simulator (ADS) [50], integrated safety assessment (ISA) [10], Monte Carlo dynamic event tree (MCDE-T) [51], analysis of dynamic accident progression tree (ADAPT) [21] and recently reactor analysis and virtual control environment (RAVEN) by INL [52–54]. All the methods differ in treatment of branching rules, stopping rules, branching generations (automatic), interfacing with existing deterministic dynamic code for physical parameters for an accident sequence, model of operator behavior. In other words, treatment of aleatory and epistemic uncertainties is different in all the methods.

DET is an event tree model in which branchings are allowed to occur at different points in time and each system can have more than two states. Initially, a fixed time interval was used for branching and then Ref. [55] introduced probabilistic branching. Reference [49] defines DETAM with five characteristic sets that define the dynamic event tree approach. These are: (i) Set of variables included in the ‘branching set.’ (ii) Set of variables defining the ‘plant state.’ (iii) Set of ‘branching rules.’ (iv) Set of sequence expansion rules. (v) Quantification tools. The ‘branching set’ is the set of variables that determine the space of possible branches. The plant state at any node in the tree is defined by the value of the variables in the branching set. The ‘branching rules’ are the rules used to determine when a branching should take place. Most straightforward branching rule could be a constant time step or based on hardware failure time, etc. The ‘sequence expansion rules’ are the rules used to limit the number of sequences, hence tree expansion. Sequence termination could be based on simulation time or reaching user-defined absorbing states or on sequence frequency falling below a user-specified lower limit. The quantitative tools are those used to compute the process variables as well as the branching frequencies.

There were various challenges in the use of DET [2, 3], but later it has been tackled in recently developed DET-based tools: first, the explosion of number of branches while allowing all possible branches in simulation, second the computational resource-intensive nature of the tools based on DET and third processing of the amount of data produced for a single initiating event. The first and second problems can be solved by increasing the time step, decreasing the probability threshold, limiting branches according to the probability cut-off or terminating the branches which exceed the user-defined number of failures or grouping of similar scenarios. The second problem can also be solved by using parallel processing [54, 56]. The main advantage of DET, which makes it a well-developed method, includes its capability to provide complete investigation of the possible accident scenarios. There is no assumption on the type of the distribution for branching, and the method provides readable results to the risk analyst. Because analysis of all the scenarios at once is difficult, many developments have been carried out in the literature for classification of similar scenarios, called scenario clustering, for more understanding of the DET results, and hence addressed the third problem of processing of the data generated for a single initiating event [57].

2.4 Dynamic Flowgraph Methodology (DFM)

The method was introduced by S. Guarro and D. Okrent, for process failure modeling in 1984. DFM accounts for the interaction between hardware and software within embedded digital system. DFM is a digraph-based technique, and it uses graphical symbols like (i) process variable node (circle), (ii) condition edge (dotted arrow), (iii) causality edge (continuous arrow), (iv) condition node (square box), (v) transition box and (vi) transfer box [2, 3, 6, 58]. A brief description of the elements of DFM is as follows. A process variable node represents a process variable which is discretized into a finite number of states. There is no criterion for discretization of the process variable, but boundary/threshold crossing and logic of the system is preferentially used for discretization. The system dynamics is represented by a cause-and-effect relationship between these states, and states are connected by causality edges. Condition edges are used to connect condition node and transfer box. The transfer box selects the transfer function according to the value present in the condition node. The transition box allows system to evolve in time by providing time lag between input and output parameters.

DFM can be used for both deductive (top to bottom) and inductive (bottom to top) tracing of fault propagation. DFM yields the prime implicants for the system.

The advantages of DFM are that it has ability to change system configuration in discrete time. This can be used for acquiring information for the possible accident sequence leading to top event. In addition to that, DFM models the system and top event separately, which is useful in case where various top event probabilities ought to be determined. Reference [6] mentions DFM method as the most preferable methodology for the reliability modeling of digital l&C systems. Application of DFM includes: control software in advanced reactors [59, 60], modeling human performance and team effects [59] and PSA modeling of a digital feedwater control system similar to an operating PWR [29].

In DFM due to the discretization of process variable, there exists inevitable discretization error. Proper discretization of each variable can be achieved by progressively refining the discretization. Hence, there is a trade-off between the accuracy of the model and the size and complexity of the model [2]. Reference [61] suggested DFM and Markov/CCMT can be used in a complementary manner, finding prime implicants in DFM, i.e., the combination of basic events necessary and sufficient to cause top event, and quantifying the prime implicants using Markov/CCMT, to achieve comprehensive modeling and evaluation of digital I&C systems. The methods to find prime implicants from the DFM include the transformation of DFM into FT [62] or into binary decision diagram (BDD) [63]. In Yadrat code, the later approach is used to evaluate DFM [58]. The other computer program based on DFM is Dymonda [61].

2.5 Event Sequence Diagram (ESD)

This is a self-explanatory diagram mainly used for understanding of accident sequences. The ESD method with dynamic capabilities is the extension of continuous event tree methodology with visual graphics. The ESD uses 6-tuple of events, conditions, gates, process parameter set, constraint and dependency rules to represent dynamic scenarios like conditions, competitions, concurrent processes (output AND gate), mutually exclusive outcomes (output OR gate), synchronization processes (input AND gate) and physical process (physical variable condition). The objective of the method is to find failure frequency and availability of the system to be in a particular state at a particular time. The ESD models the time implicitly, in a way branching in ESD occurs based on the crossing of a boundary by a process variable. The method can be utilized to predict future failure, cut sets, etc. For availability of a system in some states, all the possible states of the system need to be declared and, of course, the transition rates between them. The drawbacks of ESD include: The methodology is not a component-based approach but a scenario modeling framework. The sequence identification is analyst dependent as the sequence delineation has to be performed by the analyst and not performed automatically. In a highly dynamic system, the ESD could become very large [2]. The application of ESD includes Refs. [64, 65] for Cassini mission and transient analysis of europa reactor.

A semi-analytical solution is possible using the ESD approach [2]. The procedure is summarized as follows [2]: first, constructing the ESD using the symbolic ESD language, second deriving mathematical representation for all components in the ESD using the governing equations, third obtaining analytical approximations to the relevant process variables and finally solving the integrals analytically or numerically to obtain end-state probabilities. The job of the analytical approximations could be laborious. The semi-analytical approach reduces the problem of dimensionality of the system evolution by solving a sequence of multidimensional integral equations.

2.6 Dynamic Fault Tree (DFT)

This is the extension of the traditional fault tree with the time element. DFT was first developed by Dugan et al. [66]. One type of dynamic fault tree uses *generalized dynamic gates* for representing interaction of the basic events in time [66, 67]. For instance [67] uses the following gates: functional dependency gate (FDEP), spare gates (CSP: cold spare gate, HSP: hot spare gate, WSP: warm spare gate), priority AND gate (PAND) and sequence enforcing gate (SEQ). If $c = a$ PAND b, then c becomes true when a and b are true but only if a becomes true before b. The SEQ gate is an expression of a constraint that components can only fail in the specified order and the top event occurs when all basic events have occurred. The FDEP gate represents the generation of dependent events once the trigger event (a failure) occurs. The dependent events are multiple inputs to this gate and can be connected to inputs of other gates. This gate does not have an output. The SPARE gate is used to model different types of redundancy, i.e., cold, warm and hot. CSP means the spare component does not fail, HSP means the spare component may fail at its full rate, and warm SPARE means the spare component may fail at a rate equal to the full rate reduced by dormancy factor. However, it is not clear as to how many of these gate types are essential to model the real systems and there is no systematic analysis to our knowledge that these gates are complete to model dynamic situations.

The other type of dynamic fault tree uses *house events* with a house event matrix handling the multiple operation mode changes in time and the configuration changes with time [68]. In a house event matrix, columns describe system configuration at the time periods and rows describe the number of the house events present in the model, which timely switch on and off in accordance with the status of the modeled system. The dynamic fault tree and static FT have common disadvantages: complexity in modeling and simultaneously handling the sequential dependent failure and multiple operation mode evolution of the system. Adversely, there is no model for process evolution in these DFTs. Intuitive and convenient *reliability graph with generalized gate* (RGGG) has been improved to account for these challenges using dynamic generalized gates [69] and reliability matrix [70], respectively, where dynamic generalized gates solve problem of sequence-dependent failure and reliability matrix able to account for the limitation of multiple operation modes of dynamic fault trees. The quantification of the dynamic fault tree can be performed either by time-dependent Boolean logic, Markov models, Bayesian network [71, 72], MCS [67] or numerical integration [73].

2.7 GO-FLOW Methodology

The *GO-FLOW* [74] methodology was developed as a success-oriented methodology; however, failure states were later included in the methodology by introducing NOT gates in the method. In this method, an analyst needs to make a chart of the physical layout of the system. The chart has all possible operational states of the system. Signal lines and operators are used to make the system chart. The operators are connected by signal lines. The operator presents the functioning or failure of the component or a logical gate or a signal generator. The operator also represents transition or any Boolean logic (AND, OR, NOT). The input/output signal lines represent the process variable or time or any other information.

Using GO-FLOW method, reliability and availability can be evaluated for the complex systems. The assumptions in the modelling include constant failure rate models, two state components and the time of failure of various components are independent. The GO-FLOW methodology is useful for phased mission analysis as it provides compactness of the representation of the phased mission situation in comparison with event trees. The GO-FLOW methodology can be used as quantitative reliability evaluation of process systems after utilizing static methods (such as failure mode and effect analysis (FMEA)) for qualitative analysis for identifying failure modes [75].

There are some drawbacks in the method. The method cannot model redundant systems easily [2]. The method possesses the state space explosion problem in the case of a large number of system components. Also, the concept of the method may be hard to learn and implement for the analyst [6]. The available computer program for GO-FLOW methodology includes in Ref. [74]. The GO-FLOW methodology has been applied to nuclear systems and control system of bullet trains in Ref. [2], and passive safety system of AP1000 reactor for dynamic reliability evaluation [75].

2.8 Stochastic Hybrid Automaton (SHA)

Though a component failure can occur gradually, it is often treated as a discrete event. Following this approximation, a system under consideration can be treated as a *discrete event system (DES)*, i.e., a system whose behavior is not controlled by time but by the occurrences of events. The *SHA theory* is an extension of the theory of FSA, considering stochastic events such as failure of components and deterministic evolution of the process variables [76, 77].

Here, the procedure for dynamic risk assessment is a two-step process involving the development of an SHA model of the studied system and then MCS of the system modeled by SHA for either a predefined mission time or a stopping criterion. The SHA method is able to work with different continuous modes of the operation of the system which are generated due to control action upon threshold crossing and the deterministic or stochastic transition events due to component failure. The later events are incorporated through probability distributions, and the former is defined by ordinary differential equations in the method. The SHA model does not require preliminary simplifying hypotheses, and thus it is able to consider all types of events such as failure, repairs, maintenance, aging, control actions, demand-dependent aging and future behavior dependent on the past [4]. The SHA method shows to be a suitable tool to model large complex systems operating in a dynamic environment. The method is implemented in Scilab/Scicos environment in Ref. [77] and is applied to test cases of the heated tank system and an oven with a temperature control system. In the validation study of the SHA method for the two systems, the features of the SHA method were tested for the following cases: (1) dependencies/interaction between stochastic events and continuous variables, (2) multiple aging modes according to repairable/reconfigurable components and (3) treatment of non-temporal behavior of components such as failure on demand. References [4, 78, 79] address the application of the SHA for the dynamic scenarios of a controlled steam generator, a feedwater control system of steam generator and a decanter unit of a chemical plant, respectively. The only disadvantage of the SHA method is the size of the model and consequently the simulation time.

The available computer programs which are based on SHA are *Dynamic Reliability and Assessment (DyRelA)* [80], PythoniC-Object-Oriented Hybrid Stochastic Automata (PyCATSHOO) [81] and *Stochastic Hybrid and Non-repairable Dynamic Fault Tree Automaton (SHyFTA)* [79]. The SHA is used as input representation for PDMP in the PyCATSHOO [81] tool. Recently, PyCATSHOO has been applied to a DHR system of sodium fast reactor in Ref. [82] for dynamic PRA-informed design. The SHyFTA, is a stochastic model of dynamic reliability uses *Hybrid Basic Event (HBE)*, and it is a combination of SHA and non-repairable dynamic FT. HBE is defined as the evolution of the basic events; i.e., basic event is not simply characterized by a static cumulative distribution function of probability, but also depends on the deterministic evolution of the system. The approach of SHyFTA is based on the separation of two mutually dependent processes, the deterministic and the stochastic. Compared to DyRelA and PyCATSHOO, SHyFTA includes *reliability, availability, maintainability and safety (RAMS)* formalism and also additional feature of MATLAB framework since it has been developed in Simulink. The Simulink architecture provided in Ref. [79] is generalized and can be used for any other dynamic reliability problems.

2.9 Petri Net (PN)

This is a graphical modeling method introduced by C. A. Petri in 1966 for system modeling. PN is similar to finite state machine. PN explicitly models time using a set of elements like nodes/places, tokens, arcs and transition. The nodes, represented by circles, are used to describe system state or process variables; transition is represented by a rectangular box, describe events; and arcs are used to connect nodes/places to transition or transition to nodes/places. The token, represented by a dot in the circle, describes the current state of the system. When PNs are executed, the tokens are consumed by transition in each of the places where it is connected from and tokens are produced by transition in each of the places where it is connected to provided that while triggering all the places that are connected from are having at least one token each. Initially, firing of timed transition was having only exponential distribution or deterministic times. The times derived from constant failure rate model can be used as mapping between PN and Markov model. Then, Ref. [83] has extended the *stochastic PN (SPN)* to *generalized stochastic Petri net (GSPN)* which represents a semi-Markov process (allowing any probability density function for time sampling). The overview of PN theory and SPN is presented in Ref. [84]. In *interpreted stochastic PN*, messages (Boolean variables (0/1, true/false)) are passed along with firing of the PN. Interpreted stochastic PNs cannot be proof checked, and hence the verification process is non-intuitive and time consuming [2]. Other extensions of PNs include fuzzy PN, colored PN, timed PN, stochastic PN [85, 86].

For quantitative evaluation of dynamic system reliability using PN, process variable models can be coupled with PN and the resulting MCS can be used to perform quantitative analysis [61]. Also, PNs can be used to simulate FTs directly and hence can be readily included in PRA [87]. A standard PN with inhibitor arcs and condition places can be used to determine missing safety requirements, uncertainties in safety requirements and inconsistencies in safety requirements.

The *stochastic reward Petri net (SRPN)* has been developed for reliability analysis of communication networks by [88]. Reference [89] presented a generalization of PN to mode automata. A mode automaton is an input/output automaton with a finite number of states called modes. The mode automata can be compiled into fault trees. Mode automata are a superset of PNs with inhibitor arcs and can model everything a computer can compute [6].

PN allows explicit representation of time and is able to simulate concurrent activities, dynamic activities and time delay. It also allows studying systems that are characterized as being asynchronous, distributed, parallel, non-deterministic and/or, stochastic. The major limitation of PN is combinatorial explosion when modeling large systems.

2.10 Dynamic Bayesian Network (DBN)

As mentioned earlier, in a dynamic system, system configuration can change, which, in turn, can change the failure rate of the corresponding components. That describes the conditional dependence of failure probability on dynamics or system configurations. In these situations, Bayesian updating of conditional probability can be directly utilized. The Bayesian network (BN) is a probabilistic approach, and it explicitly represents the relationships/interactions between the system components. For graphical representation of the interactions between components and process variables of a system, BN uses nodes for describing variables/events/system components and arcs/edges connecting to nodes for describing relationships (child and parent) between the nodes. BN graphs are directed acyclic graphs. The nodes having only incoming arcs are called leaf nodes. The nodes with only outgoing arcs are independent nodes (root nodes). The root nodes are defined with marginal prior probabilities. The prior conditional probabilities are stored in *conditional probability table (CPT)* for each node with success and failure instantiations except root nodes. The probabilities of each node are updated using Bayes formula. In a complex system having n number of components, a component may not depend on all other components but only on m of them, the number of parent nodes. Hence, there are 2^m entries instead of 2^n in the conditional probability table. This reduces computational workload substantially [90]. The drawbacks of BN model include the following: BN models require prior knowledge of probability of the interactions. Further, the BN model developed is strongly expert dependent. These disadvantages can be addressed by the development of suitable algorithms for automatic construction of BN [90].

Explicit representation of time in BN is referred to as *timed Bayesian network (TBN)*. Two approaches are presented in modeling time in BN, i.e., *time-sliced approach* and *event-based approach*. Early work on the application of BNs to dynamic domains has led to formalisms known as *temporal Bayesian networks* [91], *dynamic Bayesian networks (DBNs)* [92], network of ‘dates,’ and modifiable temporal belief networks (MTBNs) are based on the time-sliced approach. The temporal node Bayesian network (TNBN) and net of irreversible events in discrete time (NIEDT) are based on the event-based approach.

The dynamic extension of BN model, i.e., DBN, is a powerful and flexible tool to model dynamic system behavior and update reliability and uncertainty analysis with life cycle data [93]. In DBN, the BN is defined at n number of time steps called n-time-sliced BN and the nodes representing component/event/variables of each time step are connected to each other using edges. The relationships between time steps are described by interconnecting edges. The state of the variables at time step n depends on time step $n - 1$, time step $n - 2$ and so on. Generally, two time slices are considered to model the system time evolution, hence the model called 2-TBN or 2-time slice temporal BN. As the state variables depend only on previous time step variables, the 2-TBN model is Markovian model. For quantification, DFTs are translated into DBNs in Refs. [71, 72]. References [90, 94–97] show recent developments in the DBN based reliability analysis.

3 Proposal of a Combination of Smart Component Methodology and Monte Carlo Simulation

In this section, we propose and briefly describe a smart component methodology (SCM) in combination with MCS as a dynamic reliability analysis method. However, the quantitative validation and any theoretical development are out of the scope of the paper. One of the requirements for a dynamic reliability modeling is a system representation. In SCM, the system representation is done through object-oriented architecture. Here, the components of a system and their interrelationship (between entities that are presented in Fig. 1) are described using abstract objects (drawn from the paradigm of object-oriented analysis and design of software systems). The objects will have attributes and rules. The attributes capture all information of the individual component, for example, the functional and reliability variables. The rules capture the behavior of the object in the system, and it will be responsible for object evolution in time. The interrelationship between components is defined in the connector, and they capture the dependencies among them. This way of representing each component or subsystem for reliability analysis by an abstract software object (with attributes, interrelation and rules) is termed smart components (SCs). Next, after the system structuring using the SC model, the SC model of the system is operated with a suitable reliability quantification algorithm. Component-based MCS is selected for reliability analysis, where MC sampling of the component states and transition time is carried out to explore the system state space and then, along with deterministic system simulation, calculate reliability measures based on the number of visits to the system failed state. This representation is perhaps the most foolproof [2]. However, the challenge lies in the development of suitable MCS algorithms or methods that translate the object-oriented representation to, for example, a DFM.

To implement the SCM, a relational database architecture is selected for representing a given system which contains all the component-as-an-object. The components would include all entities of Fig. 1, i.e., process, hardware, software and human performance. The objects contain the information of the components with the following attributes:

1. Input parameters.
2. Output parameters.
3. Parameters describing the state of the component (hardware, software, etc.).
4. Reliability parameters (failure rate, repair rate, an indicator of the reliability model, the probability of failure).

In addition to the components, the database file must contain the ‘connector’ object describing the connections between the components’ attributes which defines the hardware wiring of the system. The function of each component, manifested by the rules, may represent its functional dependence on other components or input variables. The process variables of an object are treated as a functional dependence through rules of the component. Using all this information about the system, MCS can be performed using the procedure described elsewhere.

In SCM, the analyst makes the SC model of the system and uses SC simulators for quantification. The building of the SC model is intuitive and user-friendly. Hence, the proof-of-correctness of the method is transferred to the simulation algorithm rather than the analyst. MCS handles the continuous time, multiple failure modes and aging separately, and the unification of all in one MCS algorithm for SCM is a challenge. The scalability of SCM depends on MCS algorithm, and it is not addressed here, but the presence of parallel computers and advanced computing techniques would alleviate the problem of computational intensive nature of the MC method. An advantage of SCM is that the same SC system model can be used for static reliability evaluation because the system represented using the connector and component objects can be translated to a reliability block diagram (RBD). For instance, static reliability evaluation can be done by carrying out a reachability check on the SC model. Hence, the method would be considered as a method with backward compatibility, which is expected from any dynamic reliability method.

In the next section, the dynamic reliability methods are qualitatively compared.

4 Qualitative Comparison of Various Methods

The qualitative attributes considered for this study are listed below. A 'Yes' implies the method is suitable for use in dynamic reliability assessment concerning that attribute, and 'No' implies otherwise.

A. The method provides improved accuracy compared to static reliability methods. The method takes care of the timing of occurrence of events, process variable evolution, control action, multiple failure modes, more importantly dependence of failure rate on both process parameters and time.

   I. Time declaration (continuous = Y/discrete = N).
   II. Treatment of process variable (Y/N).
   III. Multiple failure modes and aging (Y/N).

B. Burden of Proof: Equivalence of reliability model to the system model (fidelity), requirement of skillful analyst and level of approximations in the method (Y=burden of proof requirement is less).

S. Whether the method is scalable to large problems, including complexity of representation (memory) solution time (time) as a function of problem size (less than exponential growth or there is a way to handle exponential growth within the method = Y/exponential growth = N).

F. Whether the method is user-friendly or intuitive for modeling of system (Y/N).

G. Whether general-purpose software is available based on the method or suitable for being developed into one (Y/N).

I. Possibility of interfacing of the results of the analysis with FT/ET (Y/N).

H. Whether the method is able to incorporate software and human reliability model (Y/N).

Table 1 Qualitative comparison

Methods	A.I	A.II	A.III	B	S	F	G	I	H
Dynamic fault tree	Y	N	N	N	N	Y	Y	Y	N
Cell-to-cell mapping technique	Y	Y	Y	Y	N	N	N	N	N
Dynamic event tree	Y	Y	Y	Y	Y	Y	Y	Y	Y
Dynamic flowgraph methodology	Y	Y	Y	N	N	N	Y	Y	Y
GO-FLOW methodology	Y	N	Y	N	N	N	N	N	Y
Petri nets	Y	N	Y	Y	Y	N	Y	N	Y
Stochastic hybrid automaton	Y	Y	Y	Y	Y	N	Y	N	Y
Event sequence diagram	Y	Y	N	N	N	Y	N	Y	Y
Dynamic Bayesian network	Y	N	Y	N	N	Y	Y	N	Y
Piecewise deterministic Markov processes	Y	Y	Y	Y	N	N	N	N	N
Smart component methodology	Y	Y	Y	Y	Y	Y	Y	Y	Y
Direct Monte Carlo simulation	Y	Y	Y	Y	N	N	N	Y	Y

From Table 1, the most favorable method is the method with the most Y's. That includes DET, SHA, SCM and direct MCS. In DET, the deductive analysis is difficult (top event to basic event). SHA theory is then the second best method in the list with two disadvantages of not being able to interface the result with fault tree methodology and being less user-friendly and intuitive. Though the direct MC method is scalable and in principle can model anything, it lacks a system-to-simulation model translation mechanism. Hence, it is not readily usable by non-experts. This problem is precisely addressed in the SCM and hence is promising regarding most of the attributes.

We believe that one of the principal reasons for non-availability of a general-purpose tool for dynamic reliability assessment is the disconnect between representation schemes and simulation power. Many of the easy and intuitive methods do not scale up. Similarly, MCS which in principle can solve any problem requires a general-purpose representation scheme and interface. With this perspective, when the methods are evaluated for the attributes B, S, F, G and I, smart component method emerges as a very likely candidate for a general-purpose dynamic reliability analysis method.

5 Conclusion

We have presented a brief review of some available dynamic reliability methods with a view of selecting and developing the best among them for use in reliability analysis of dynamic systems. The attributes required for an ideal dynamic method are elucidated, and the reviewed methods are inter-compared. In the inter-comparison, we emphasize the general applicability and suitability of the methods for generic tool development, as these attributes mainly have contributed to the lack of widespread use of dynamic reliability methodology. Based on the current review, SCM is identified to have excellent potential for the widespread application, while capable of faithfully modeling dynamic scenarios including software and human reliability aspects. The framework provides the structure for the dynamic system representation, while MCS is the key computational method.

6 Future Developments

A framework for the implementation of the SCM, utilizing object-oriented design and relational database concepts, is in progress. The MCS algorithms need to be explored for unifying various MC reliability models and at the same time acceleration of results. Completeness of MC state space exploration is to be studied for verifying the completeness of results since the size of state space in dynamic systems is huge. Demonstration of application of the SCM for small- to an industrial-scale system is to be studied.

Acknowledgements The authors thank Martin B. Sattison and Diego Mandelli for their valuable feedback on the draft manuscript. The authors thank Director, Reactor Design Group, IGCAR, for the encouragement and support for completing the work. First, the author thanks the Board of Research in Nuclear Studies, Mumbai, India, and Department of Atomic Energy, India, for supporting through DGFS-Ph.D. fellowship.

References

1. J. Devooght, Dynamic reliability. Adv. Nucl. Sci. Technol. **25** (1997)
2. P.E. Labeau, C. Smidts, S. Swaminathan, Dynamic reliability: towards an integrated platform for probabilistic risk assessment. Reliab. Eng. Syst. Saf. **68**, 219–254 (2000)
3. T. Aldemir, A survey of dynamic methodologies for probabilistic safety assessment of nuclear power plants. Ann. Nucl. Energy **52**, 113–124 (2013)
4. G. Babykina, N. Brînzei, J.-F. Aubry, G. Deleuze, Modeling and simulation of a controlled steam generator in the context of dynamic reliability using a stochastic hybrid automaton. Reliab. Eng. Syst. Saf. **152**, 115–136 (2016)
5. N.O. Siu, Risk assessment for dynamic systems: an overview. Reliab. Eng. Syst. Saf. **43**(1), 43–73 (1994)
6. T. Aldemir, D. Miller, M.P. Stovsky, J. Kirschenbaum, P. Bucci, A.W. Fentiman, L.T. Mangan, Current State of Reliability Modeling Methodologies for Digital Systems and Their Acceptance Criteria for Nuclear Power Plant Assessments (NUREG/CR-6901), Technical report (2006)
7. T. Aldemir, M. Stovsky, J. Kirschenbaum, D. Mandelli, P. Bucci, L.A. Mangan, D. Miller, X. Sun, E. Ekici, S. Guarro, M. Yau, B. Johnson, C. Elks, S. Arndt, Dynamic Reliability Modeling of Digital Instrumentation and Control Systems for Nuclear Reactor Probabilistic Risk Assessments (NUREG/CR-6942), Technical report (2007)
8. T. Aldemir, S. Guarro, J. Kirschenbaum, D. Mandelli, L.A. Mangan, P. Bucci, M. Yau, B. Johnson, C. Elks, E. Ekici, M. Stovsky, D. Miller, X. Sun, S. Arndt, Q. Nguyen, D.J, A

Benchmark Implementation of Two Dynamic Methodologies for the Reliability Modeling of Digital (NUREG/CR-6985), Technical report (2009)

9. T.L. Chu, G. Martinez-Guridi, M. Yue, J. Lehner, P. Samanta, Traditional Probabilistic Risk Assessment methods for digital system (NUREG/CR-6962), Technical report (2008)
10. J.M. Izquierdo, E. Melendez, J. Devooght, Relationship between probabilistic dynamics and event trees. Reliab. Eng. Syst. Saf. **52**, 197–209 (1996)
11. J. Devooght, C. Smidts, Probabilistic reactor dynamics-I: the theory of continuous event trees. Nucl. Sci. Eng. **240**, 229–240 (1992)
12. C. Smidts, J. Devooght, Probabilistic reactor dynamics-II: a Monte Carlo study of a fast reactor transient. Nucl. Sci. Eng. **111**, 241–256 (1992)
13. J. Devooght, C. Smidts, Probabilistic reactor dynamics-III. A framework for time-dependent interaction between operator and reactor during a transient involving human error. Nucl. Sci. Eng. **3**(112), 101–113 (1992)
14. T. Aldemir, Computer-assisted markov failure modeling of process control systems. IEEE Trans. Reliab. 36(1), 133–144 (1987)
15. J. Devooght, C. Smidts, Probabilistic dynamics as a tool for dynamic PSA. Reliab. Eng. Syst. Saf. **8320**(95), 185–196 (1996)
16. T. Aldemir, Some measure theoretic issues in probabilistic dynamics. Nucl. Sci. Eng. **155**, 497–507 (2007)
17. M. Belhadj, T. Aldemir, The cell to cell mapping technique and Chapman-Kolmogorov representation of system dynamics. J. Sound Vib. **181**(4), 687–707 (1995)
18. C. Smidts, Probabilistic dynamics: a comparison between continuous event trees and a discrete event tree model. Reliab. Eng. Syst. Saf. **44**, 189–206 (1994)
19. M. Hassan, T. Aldemir, A data base oriented dynamic methodology for the failure analysis of closed loop control systems in process plants. Reliab. Eng. Syst. Saf. **27**(3), 275–322 (1990)
20. T. Aldemir, M. Belhadj, L. Dinca, Process reliability and safety under uncertainties. Reliab. Eng. Syst. Saf. **52**(3 SPEC. ISS.), 211–225 (1996)
21. A. Hakobyan, R. Denning, T. Aldemir, S. Dunagan, D. Kunsman, A Methodology for Generating Dynamic Accident Progression Event Trees for Level-2 PRA, in *PHYSOR-2006, ANS Topical Meeting on Reactor Physics*, Vancouver, BC, Canada (2006), B034 1–9
22. P. Bucci, J. Kirschenbaum, L.A. Mangan, T. Aldemir, C. Smith, T. Wood, Construction of event-tree/fault-tree models from a Markov approach to dynamic system reliability **93**(2008), 1616–1627 (2015)
23. J. Yang, T. Aldemir, An algorithm for the computationally efficient deductive implementation of the Markov/cell-to-cell-mapping technique for risk significant scenario identification. Reliab. Eng. Syst. Saf. **145**, 1–8 (2016)
24. M. Belhadj, T. Aldemir, Some computational improvements in process system reliability and safety analysis using dynamic methodologies. Reliab. Eng. Syst. Saf. **52**, 339–347 (1996)
25. P.E. Labeau, Modeling PSA problems-II: a cell-to-cell transport theory approach. Nucl. Sci. Eng. **150**, 140–154 (2005)
26. B. Tombuyses, T. Aldemir, Continuous cell-to-cell mapping. J. Sound Vib. **202**(3), 395–415 (1997)
27. B. Tombuyses, J. Devooght, Solving Markovian systems of O.D.E. for availability and reliability calculations. Reliab. Eng. Syst. Saf. **48**(1), 47–55 (1995)
28. M. Belhadj, M. Hassan, T. Aldemir, On the need for dynamic methodologies in risk and reliability studies. Reliab. Eng. Syst. Saf. **38**(3), 219–236 (1992)
29. T. Aldemir, S. Guarro, D. Mandelli, J. Kirschenbaum, L.A. Mangan, P. Bucci, M. Yau, E. Ekici, D. Miller, X. Sun, S. Arndt, Probabilistic risk assessment modeling of digital instrumentation and control systems using two dynamic methodologies. Reliab. Eng. Syst. Saf. **95**(10), 1011–1039 (2010)
30. J. Kirschenbaum, P. Bucci, M. Stovsky, D. Mandelli, T. Aldemir, M. Yau, S. Guarro, E. Ekici, S.A. Arndt, A benchmark systems for comparing reliability modeling approaches for digital instrumentation and control systems. Nucl. Technol. **165**, 53–95 (2009)

31. D.L. Deoss, N.O. Siu, A simulation model for dynamic system availability analysis, Technical report (1989)
32. E.E. Lewis, F. Boehm, Monte Carlo simulation of Markov unreliability models. Nucl. Eng. Des. **77**, 49–62 (1984)
33. M. Marseguerra, E. Zio, Optimizing maintenance and repair policies via a combination of genetic algorithms and Monte Carlo simulation. Reliab. Eng. Syst. Saf. **68**, 69–83 (2000)
34. M. Marseguerra, E. Zio, Nonlinear Monte Carlo reliability analysis with biasing towards top event. Reliab. Eng. Syst. Saf. **40**(1), 31–42 (1993)
35. P.E. Labeau, E. Zio, Procedures of Monte Carlo transport simulation for applications in system engineering. Reliab. Eng. Syst. Saf. **77**, 217–228 (2002)
36. M. Marseguerra, E. Zio, Monte Carlo biasing in reliability calculation with deterministic repair times. Ann. Nucl. Energy **27**(2000), 639–648 (1999)
37. H. Cancela, P.L. Ecuyer, M. Lee, G. Rubino, B. Tuffin, Analysis and improvements of path-based methods for Monte Carlo reliability evaluation of static models. Improv. Monte Carlo Reliab. Eval. static Model. (Chap. 1) (2009), 1–20
38. M. Grigoriu, G. Samorodnitsky, Reliability of dynamic systems in random environment by extreme value theory. Probabilistic Eng. Mech. **38**, 54–69 (2014)
39. P.E. Labeau, E. Zio, The cell-to-boundary method in the frame of memorization-based Monte Carlo algorithms. Math. Comput. Simul. **47**, 347–360 (1998)
40. P.E. Labeau, A Monte Carlo estimation of the marginal distributions in a problem of probabilistic dynamics. Reliab. Eng. Syst. Saf. **52**(1), 65–75 (1996)
41. M. Marseguerra, E. Zio, J. Devooght, P.E. Labeau, A concept paper on dynamic reliability via Monte Carlo simulation. Math. Comput. Simul. **47**, 371–382 (1998)
42. P.E. Labeau, Monte Carlo estimation of generalized unreliability in probabilistic dynamics-I: application to a pressurized water reactor pressurizer. Nucl. Sci. Eng. **126**(2), 131–145 (1997)
43. M. Marseguerra, E. Zio, The cell-to-boundary method in Monte Carlo-based dynamic PSA. Reliab. Eng. Syst. Saf. **48** (1995)
44. J.M. Lanore, C. Villeroux, F. Bouscatie, N. Maigret, Progress in methodology for probabilistic assessment of accidents: timing of accident sequences, in *ANS - ENS - International ANC/ENS Topical Meeting Probabilistic Risk Assessment*, Port-Chester, NY, USA (1981)
45. Y.H. Lin, Y.-F. Li, Dynamic reliability models for multiple dependent competing degradation processes, in *Safety and Reliability: Methodology and Applications—Proceedings of the European Safety and Reliability Conference ESREL 2014*, pp. 775–782 (1984)
46. H. Zhang, F. Dufour, Y. Dutuit, K. Gonzalez, Piecewise deterministic Markov processes and dynamic reliability. Proc. Inst. Mech. Eng. Part O J. Risk Reliab. **222**(4), 545–551 (2008)
47. A. Amendola, G. Reina, Event sequences and consequence spectrum: a methodology for probabilistic transient analysis. Nucl. Sci. Eng. **77**, 297–315 (1981)
48. G. Cojazzi, The DYLAM approach for the dynamic reliability analysis of systems. Reliab. Eng. Syst. Saf. **52**, 279–296 (1996)
49. C.G. Acosta, N.O. Siu, *Dynamic Event Tree Analysis Method* (DETAM) for accident Sequence Analysis, Technical report, 1991)
50. K.-S. Hsueh, A. Mosleh, The development and application of the accident dynamic simulator for dynamic probabilistic risk assessment of nuclear power plants. Reliab. Eng. Syst. Saf. **52**, 297–314 (1996)
51. K. Martina, P. Jörg, MCDET: a probabilistic dynamics method combining monte carlo simulation with the discrete dynamic event tree approach. Nucl. Sci. Eng. **153**(2), 137–156 (2006)
52. A. Alfonsi, C. Rabiti, D. Mandelli, J.J. Cogliati, R.A. Kinoshita, A. Naviglio, S.C. Columbia, Erin, F. Garrick, A. Hughes, I. Maracor Technical Services, et al., Dynamic Event Tree analysis through RAVEN, in *International Topical Meeting Probabilistic Safety Assessment Analysis 2013, PSA 2013*, 3 (October 2015) (2013) pp. 1697–1709
53. A. Alfonsi, C. Rabiti, D. Mandelli, J. Cogliati, C. Wang, P.W. Talbot, D.P. Maljovec, C. Smith, *RAVEN Theory Manual* (Technical Report March, Idaho National Laboratory, 2016)
54. A. Alfonsi, C. Rabiti, J. Cogliati, D. Mandelli, S. Sen, R. Kinoshita, C. Wang, P.W. Talbot, D. Maljovec, A. Slaughter, C. Smith, *Dynamic Event Tree Advancements and Control Logic Improvements* (Technical Report September, Idaho National Laboratory, 2015)

55. A. Amendola, Accident sequence dynamic simulation versus event trees. Reliab. Eng. Syst. Saf. **22**(1–4), 3–25 (1988)
56. A. Hakobyan, T. Aldemir, R. Denning, S. Dunagan, D. Kunsman, B. Rutt, U. Catalyurek, Dynamic generation of accident progression event trees **238**, 3457–3467 (2008)
57. D. Mandelli, A. Yilmaz, T. Aldemir, K. Metzroth, R. Denning, Scenario clustering and dynamic probabilistic risk assessment. Reliab. Eng. Syst. Saf. **115**, 146–160 (2013)
58. K. Björkman, Solving dynamic flowgraph methodology models using binary decision diagram-s. Reliab. Eng. Syst. Saf. **111**, 206–216 (2013)
59. A. Milixi, R.J. Mulvihill, S. Guarro, J.J. Persensky, *Extending the Dynamic Flowgraph Methodology (DFM) to Model Human Performance and Team Effects*, US NRC (2001)
60. M. Yau, S. Guarro, G. Apostolakis, Demonstration of the dynamic flowgraph methodology using the Titan II space launch vehicle digital flight control system. Reliab. Eng. Syst. Saf. **49**(3), 335–353 (1995)
61. K. Björkman *Digital Automation System Reliability Analysis - Literature survey* (Technical report, 2010)
62. S. Guarro, D. Okrent, The logic flowgraph: a new approach to process failure modeling and diagnosis for disturbance analysis applications. Nucl. Technol. **67**, 348–359 (1984)
63. T. Tyrväinen, Prime implicants in dynamic reliability analysis. Reliab. Eng. Syst. Saf. **146**, 39–46 (2016)
64. S. Swaminathan, C. Smidts, The event sequence diagram framework for dynamic probabilistic risk assessment. Reliab. Eng. Syst. Saf. **63**, 224 (1999)
65. S. Swaminathan, J.-Y. Van-Halle, C. Smidts, A. Mosleh, S. Bell, K. Rudolph, R.J. Mulvihill, B. Bream, The Cassini Mission probabilistic risk analysis: comparison of two probabilistic dynamic methodologies. Reliab. Eng. Syst. Saf. **58**(1), 1–14 (1997)
66. J.B. Dugan, S.J. Bavuso, M.A. Boyd, Dynamic fault-tree models for fault-tolerant computer systems. IEEE Trans. Reliab. **41**(3), 363–377 (1992)
67. K.D. Rao, V. Gopika, V. Sanyasi Rao, H. Kushwaha, A. Verma, A. Srividya, Dynamic fault tree analysis using Monte Carlo simulation in probabilistic safety assessment. Reliab. Eng. Syst. Saf. **94**(4), 872–883 (2009)
68. M. Čepin, B. Mavko, A dynamic fault tree. Reliab. Eng. Syst. Saf. **75**(1), 83–91 (2002)
69. S.K. Shin, P.H. Seong, A quantitative assessment method for safety-critical signal generation failures in nuclear power plants considering dynamic dependencies. Ann. Nucl. Energy **38**(2–3), 269–278 (2011)
70. S.K. Shin, Y.G. No, P.H. Seong, Improvement of the reliability graph with general gates to analyze the reliability of dynamic systems that have various operation modes. Nucl. Eng. Technol. **48**(2), 386–403 (2016)
71. S. Montani, L. Portinale, A. Bobbio, D. Codetta-raiteri, Radyban: a tool for reliability analysis of dynamic fault trees through conversion into dynamic Bayesian networks. Reliab. Eng. Syst. Saf. **93**(7), 922–932 (2008)
72. A. Bobbio, L. Portinale, M. Minichino, E. Ciancamerla, Improving the analysis of dependable systems by mapping fault trees into Bayesian networks. Reliab. Eng. Syst. Saf. **71**(3), 249–260 (2001)
73. S.V. Amari, G. Dill, E. Howald, A new approach to solve dynamic fault trees. Reliab. Maintainab. Symp. 374–379 (2003)
74. T. Matsuoka, M. Kobayashi, GO-FLOW: a new reliability analysis methodology. Nucl. Sci. Eng. **98**, 64–78 (1988)
75. M. Hashim, H. Yoshikawa, T. Matsuoka, M. Yang, Quantitative dynamic reliability evaluation of AP1000 passive safety systems by using FMEA and GO-FLOW methodology. J. Nucl. Sci. Technol. **51**(4), 526–542 (2014)
76. J.-F. Aubry, N. Brînzei, *Systems Dependability Assessment: Modeling with Graphs and Finite State Automata*, risk management edition (Wiley, New York, 2015)
77. G.A.P. Castañeda, J.-F. Aubry, N. Brinzel, Stochastic hybrid automata model for dynamic reliability assessment. IMechE 28–41 (2015)

78. G. Babykina, N. Brînzei, J.-F. Aubry, G. Deleuze, Modelling a feed-water control system of a steam generator in the framework of the dynamic reliability, *ESREL-2013* (2014), pp. 3099–3107
79. F. Chiacchio, D.D. Urso, L. Compagno, M. Pennisi, F. Pappalardo, G. Manno, SHyFTA, a stochastic hybrid fault tree automaton for the modelling and simulation of dynamic reliability problems. Expert Syst. Appl. **47**, 42–57 (2016)
80. G.A.P. Castañeda, J.-F. Aubry, N. Brînzei, DyRelA (Dynamic Reliability and Assessment), pp. 39–40 (2010)
81. H. Chraibi, Dynamic reliability modeling and assessment with PyCATSHOO: application to a test case, in *PSAM* (2013)
82. V. Rychkov, H. Chraibi, Dynamic probabilistic risk assessment at a design stage for a sodium fast reactor, in *IAEA-CN245-042*, Yekaterinburg (2017), pp. 1–8
83. A. Marson, *Modeling with Generalized Stochastic Petri Nets* (Wiley, New York, 1995)
84. C. Cordier, M. Fayot, A. Leroy, A. Petit, Integration of process simulations in availability studies. Reliab. Eng. Syst. Saf. 8320 (96)
85. L. Gomes, J.P. Barros, Addition of fault detection capabilities in automation applications using Petri nets, in (IEEE, New York, 2004), pp. 645–650
86. L. Zhenjuan, P. Bo, L. Hongguang, Batch process fault diagnosis based on fuzzy petri nets, in *Proceedings of the First International Conference Innovations and Computer Information Control* (2006)
87. T.S. Liu, S.B. Chiou, The application of Petri nets to failure analysis. Reliab. Eng. Syst. Saf. **57**, 129–142 (1997)
88. M. Balakrishman, K. Trivedi, Stochastic Petri nets for reliability analysis of communication network applications with alternate routing. Reliab. Eng. Syst. Saf. (53), 243–259 (1996)
89. A. Rauzy, Mode automata and their compilation into fault trees. Reliab. Eng. Syst. Saf. (78), 1–12 (2002)
90. O. Doguc, J.E. Ramirez-Marquez, A generic method for estimating system reliability using Bayesian networks. Reliab. Eng. Syst. Saf. **94**(2), 542–550 (2009)
91. K. Kanazawa, Reasoning about Time and Probability Keiji Kanazawa, Ph.D. thesis, Brown University (1992)
92. A.E. Nicholson, J.M. Brady, Dynamic belief networks for discrete monitoring. IEEE Trans. Syst. Man. Cybern. **24**(11), 1593–1610 (1994)
93. J. Zhu, M. Collette, A dynamic discretization method for reliability inference in dynamic Bayesian networks. Reliab. Eng. Syst. Saf. **138**, 242–252 (2015)
94. M. Kalantarnia, F. Khan, K. Hawboldt, Dynamic risk assessment using failure assessment and Bayesian theory. J. Loss Prev. Process Ind. **22**(5), 600–606 (2009)
95. H. Boudali, J.B. Dugan, A discrete-time Bayesian network reliability modeling and analysis framework. Reliab. Eng. Syst. Saf. **87**(3), 337–349 (2005)
96. P. Weber, L. Jouffe, Complex system reliability modelling with Dynamic Object Oriented Bayesian Networks (DOOBN). Reliab. Eng. Syst. Saf. **91**(2), 149–162 (2006)
97. A. Ben Salem, A. Muller, P. Weber, Dynamic Bayesian networks in system reliability analysis, in *6th IFAC Symposium Fault Detection, Supervision and Safety Technical Processes*, pp. 481–486