A BENCHMARK SYSTEM FOR COMPARING RELIABILITY MODELING APPROACHES FOR DIGITAL INSTRUMENTATION AND CONTROL SYSTEMS

NUCLEAR PLANT OPERATIONS AND CONTROL

KEYWORDS: *Markov, dynamic flowgraph methodology, dynamic PRA*

JASON KIRSCHENBAUM and PAOLO BUCCI *The Ohio State University*
Department of Computer Science and Engineering, 395 Dreese Laboratories
2015 Neil Avenue, Columbus, Ohio 43210

MICHAEL STOVSKY, DIEGO MANDELLI, and TUNC ALDEMIR*
The Ohio State University, Nuclear Engineering Program, 427 Scott Laboratory
201 West 19th Avenue, Columbus, Ohio 43102

MICHAEL YAU and SERGIO GUARRO *ASCA, Inc.*
1720 S. Catalina Avenue, Suite 220, Redondo Beach, California 90277

EYLEM EKICI *The Ohio State University*
Department of Electrical and Computer Engineering, 320 Dreese Laboratories
2015 Neil Avenue, Columbus, Ohio 43210

STEVEN A. ARNDT *U.S. Nuclear Regulatory Commission*
Office of Nuclear Regulatory Research, Washington, D.C. 20555

Received August 1, 2007
Accepted for Publication May 29, 2008

There is an accelerating trend to upgrade and replace nuclear power plant analog instrumentation and control systems with digital systems. While various methodologies are available for the reliability modeling of these systems for plant probabilistic risk assessments, there is no benchmark system that can be used as the basis for methodology comparison. A system representative of the steam generator feedwater control systems in pressurized water reactors is proposed for such a comparison. Dynamic reliability modeling of the benchmark system for an example initiating event is illustrated using the Markov/cell-to-cell mapping technique and dynamic flowgraph methodologies.

I. INTRODUCTION

Nuclear power plants are in the process of replacing and upgrading aging and obsolete instrumentation and control (I&C) systems. Most of these replacements involve transitions from analog to digital technology. Digital systems differ from analog systems mainly due to the presence of software and firmware. While the 1995 U.S. Nuclear Regulatory Commission (NRC) probabilistic risk assessment (PRA) policy statement[1] encourages the increased use of PRA and associated analyses in all regulatory matters to the extent supported by the state of the art in PRA and the data, there are presently no universally accepted methods for modeling digital systems for the purpose of identifying software-related failure modes and their system-level effects and of integrating this

*E-mail: aldemir.1@osu.edu

information into current-generation PRAs. The recently published NUREG/CR-6901 (Ref. 2) has identified, among the methodologies with potential for such modeling, the Dynamic Flowgraph Methodology (DFM) and the Markov/Cell-to-Cell Mapping Technique (Markov/CCMT). However, NUREG/CR-6901 also concluded that the lack of a realistic benchmark (a known model of a system similar to those of operating nuclear power plants supported by operational data) against which methodologies can be evaluated poses an obstacle to an objective comparison of their advantages and limitations.

A recent study has identified the desirable features of such a benchmark system.[3] The objective of this paper is to propose a benchmark system that has most of these features (Sec. III) and to illustrate how the prime implicants for the top events can be obtained using the DFM and/or Markov/CCMT modeling approaches (Sec. IV). In reliability engineering, prime implicants are counterparts of minimal cut sets for multivalued logic structures and for those binary logic structures, such as noncoherent fault trees (FTs), which may have NOT, NAND, NOR, and XOR gates as well as the AND and OR gates.[4] Both of these classes of logic structures are often relevant to systems where the timing of failures may affect the nature and frequency of the top events. Section II describes the differences between analog and digital I&C systems with regard to their reliability modeling and explains why dynamic PRA methodologies may be needed for the reliability modeling of digital I&C systems. Dynamic PRA methodologies are those that explicitly account for the time element in system evolution to represent the possible statistical dependence between failure events due to either through[2] (a) indirect interaction through the monitored/controlled process (type I) or (b) direct interaction though hardware/software/firmware (type II). A recent review of dynamic PRA methodologies relevant to digital I&C systems can be found in NUREG/CR-6901 (Ref. 2). Section VIII gives the conclusions of the study carried out with the application of DFM and Markov/CCMT to the benchmark system discussed in Sec. III.

II. ANALOG VERSUS DIGITAL I&C SYSTEMS

NUREG/CR-6901 has identified a number of special characteristics of digital I&C systems, which differ from their analog counterparts with specific regard to their functional and reliability modeling. These characteristics can be grouped into categories A through D:

Category A—Complexity characteristics related to the extended capability and functionality

Characteristic A.1. There may be complex interactions between the components of the digital I&C system and the process physics or environment (type I interactions), as well as among the components themselves (type II interactions), which may lead to potentially significant dependencies between failures events.[5]

Characteristic A.2. A digital controller not only reacts to data but also can anticipate the state of the system.

Category B—Performance characteristics related to the nature of digital processes and devices

Characteristic B.1. Artifacts[a] *and aliasing*[b] *may be introduced if the sampling rate is too low for the application*[6] *or the binary approximation introduces significant round-off or truncation errors*, since digital systems operate in discrete time steps and use binary approximation of real numbers.

Characteristic B.2. Digital I&C systems rely on sequential circuits that have memory. Consequently, digital I&C system outputs may be a function of system history, as well as the rate of progress of the tasks.

Characteristic B.3. Digital systems may have a much smaller operating environment temperature range than analog counterparts and may be affected differently from analog systems by external stressors such as electromagnetic/radio-frequency interference, temperature, pressure, vibration, and radiation.

Category C—Failure mode characteristics

Characteristic C.1. The failure mechanisms of digital systems may not be well understood and defined.[7] Errors in design and software implementation can cause the digital system, which appeared to be functioning correctly, to fail suddenly because of some specific input received.

Characteristic C.2. Tasks may compete for a digital controller's resources. This competition requires coordination between the tasks and may lead to problems such as deadlock and starvation.

Characteristic C.3. New failure modes can be introduced in digital I&C systems because of a higher degree of data sharing and communication. Communication protocols may introduce dependencies between different systems such that if a device fails in a way that introduces invalid data as input to other devices via the communication links, the invalid data subsequently may cause all other systems using that input resource to fail. Similarly, multitasking within the same communication link may also introduce new failure dependencies due to protocol interdependencies.[c]

---

[a] An artifact is any perceived distortion or other datum error caused by the instrument of observation.

[b] Aliasing occurs when two different continuous signals become the same when sampled by a particular device.

[c] See Information Notice 2007-15 (Ref. 8).

Characteristic C.4. Software may be able to mask intermittent failures in hardware[9] *and has the ability to introduce corrective actions or mitigate failed hardware through fault tolerance or fault recovery.*[10]

Characteristic C.5. Digital I&C systems may be more vulnerable to common-cause failure since they include software whose failure may affect multiple functions. They also share data transmissions, functions, and process equipment to a greater degree than analog systems.

Category D—Reliability and test characteristics

Characteristic D.1. The digital I&C firmware/software reliability cannot be accurately modeled using a bathtub curve approach.[11] Software defects may remain hidden for long periods after a product has been in general use, and failures may occur without any advance warning when a particular execution path is exercised.[7]

Characteristic D.2. The firmware and software components of digital I&C system do not demonstrate any wear characteristics in the conventional sense. Consequently, these elements of digital systems do not respond to accelerated life testing and stress testing.

Characteristic D.3. Software is not a physical entity whose own intrinsic nature bounds the potential failure domains. Thus, testing alone is not sufficient to verify that software is complete and correct.[7]

Particularly because of the characteristics under category D, some key assumptions underlying traditional approaches to the reliability modeling of ordinary hardware systems are no longer valid when modeling digital and software-intensive systems. This in turn leaves open the question concerning the validity of those digital I&C reliability models that treat firmware and software as a largely invisible element of the hardware that encapsulates it.[2] As to the question of what type of models may be needed to cover the potentially important failure mode characteristics of digital I&C systems, i.e., category C, as well as the characteristics under categories A and B, all provide reasons to believe that in general, the functional and dynamic complexity of these systems calls for the use of dynamic modeling techniques in order to achieve an adequate level of fidelity to their actually stochastic performance.

To support the identification of a suitable modeling approach, NUREG/CR-6901 has identified a set of requirements for a methodology that can be used for the reliability modeling of digital I&C systems[2]:

Requirement 1. The model must be able to predict future failures well and not rely only on operation or experience or testing. For example, "black-box" models (e.g., Ref. 12) do not satisfy this requirement since these models may not be able to predict the consequences of event sequences that were not part of the training data. Similarly, failure data based on operational experience may not be able to account for the aging of the digital I&C system hardware and inputs into the system that fall outside the design domain of the digital I&C system.

Requirement 2. The model must account for the relevant features of the system under consideration. If the digital I&C system is used strictly for data collection with no processing of data or decision making, then the conventional event-tree (ET)/FT approach can often be satisfactory. However, data collection from sensors may require analog-to-digital conversion, which may introduce errors or artifacts if the sampling rate is not sufficiently high[6] or through the failure to use proper antialiasing techniques. If sequence dependent failure modes exist, a state-based technique (e.g., Ref. 13) may need to be used. Extensive interaction of the digital I&C system with process physics may require more complicated modeling procedures [such as continuous ET (CET) methodology[14] or Markov modeling through the cell-to-cell mapping technique[15] (CCMT)].

Requirement 3. The model must make valid and plausible assumptions. The conventional ET/FT approach assumes that faults occurring in system components propagate instantaneously throughout the system. There is evidence that such an assumption leads to overestimation of top event frequencies in control systems with more than one failure mode.[16,17] There is also evidence that the assumption (along with qualitative representation of the process physics in the ET/FT approach) may lead to incomplete identification of the scenarios leading to the top event[17,18] and incorrect quantification of the statistical importance of component failures with respect to the top event.[19]

Requirement 4. The model must quantitatively be able to represent dependencies between failure events accurately (see characteristics A.1, B.3, C.1, and C.3).

Requirement 5. The model must be designed so it is not difficult for an analyst to learn the concepts and it is not difficult to implement. For example, while the CET methodology[14] and Markov/CCMT (Ref. 15) satisfy all the requirements above, it is difficult for the analyst to learn the concepts and difficult to implement because of the current unavailability of tools to make their internals transparent to the user.

Requirement 6. The data used in the quantification process must be credible to a significant portion of the technical community. There is little operational experience with digital I&C system and field data. Consequently, most of the data to be used in the reliability modeling of digital I&C systems need to be generated or estimated from generic digital processor data. Techniques have been proposed to accomplish this need (e.g., Refs. 20, 21, and 22). However, data generation may take an unreasonable amount of time to create, run, and justify its correctness, and the test cases used might not be

representative of real workloads. If software is treated as a separate entity, the validity of the software failure data estimated may be debatable.

Requirement 7. The model must be able to differentiate between a state that fails one safety check and those that fail multiple ones (see characteristic B.3).

Requirement 8. The model must be able to differentiate between faults that cause function failures and intermittent failures (see characteristic C.4).

Requirement 9. The model must have the ability to provide relevant information to users, including cut sets, probabilities of failure, and uncertainties associated with the results. For example, while Monte Carlo simulation[23] satisfies requirements 1 through 4, there is no generic procedure to obtain the cut sets from the results.

Requirement 10. The methodology must be able to model the digital I&C system portions of accident scenarios to such a level of detail and completeness that nondigital I&C system portions of the scenario can be properly analyzed and practical decisions can be formulated and analyzed. This requirement, along with requirement 9, is relevant to the incorporation of the reliability model for the digital I&C system into an existing PRA to assess potential impacts on core damage and early release frequencies because of a conversion from analog I&C to digital. The incorporation process needs to assure proper linking between digital I&C system constituents and the other plant systems.

Requirement 11. The model should not require highly time-dependent or continuous plant state information. For example, DFM (Ref. 13) and Markov/CCMT (Ref. 15) can use a wide spectrum of models describing system evolution in time, from input-output data in tabular form to complex computer codes. CET (Ref. 14), on the other hand, requires the constitutive equations describing the system dynamics. This requirement is also relevant to the incorporation of the reliability model for the digital I&C system into an existing PRA.

Using subjective criteria based on reported experience with dynamic methodologies, NUREG/CR-6901 (Ref. 2) has identified DFM and Markov/CCMT as the methodologies that rank as the top two with the most positive features and the least negative or uncertain features when evaluated against the requirements for the reliability modeling of digital I&C systems. NUREG/CR-6901 also concluded that benchmark systems need to be defined to allow objective assessment of suitability of the methodologies proposed for the reliability modeling of digital I&C systems using a common set of hardware/software/firmware states and state transition data.

A recent study has delineated the desirable features of such a benchmark system in view of the differences between the analog and digital I&C systems listed above and the current state of digital technology.[3] These features are listed in Table I. With respect to the terminology used in Table I, "loosely control coupled" (LCC) systems are those with (only) potential type I coupling between failure events. "Tightly control coupled" (TCC) systems are those with both potential type I and type II coupling between failure events. Real time constraints (Table I LCC feature 3) refer to the time spent in processing data by the digital I&C system. Interrupts (Table I LCC feature 5) suspend the processor's current execution stream when a particular event of interest occurs, such as the availability of data on an input device. Interrupts are used frequently to facilitate communication between a processor and a much slower peripheral device such as a disk or a means of ensuring tasks are performed at regular intervals. A watchdog timer (Table I LCC feature 9) works by requiring software to signal the watchdog timer at predefined intervals. If the timer is not signaled, a fault is assumed to have occurred, and the watchdog performs some mitigating action (e.g., reboot a processor, turn off motors, open valves, switch controllers, notify other systems). Data races (Table I TCC feature 4a) refer to the situation in which the order of events executed determines the value of the data stored in shared memory. Deadlocks and starvation (Table I TCC feature 4b) may occur if more that one device is competing for the same shared resource. Finally, Byzantine failures (Table I TCC feature 6) imply that the system may do anything, including malicious behavior. Systems with Byzantine failures may also collude in performing malicious behavior.[24]

III. THE BENCHMARK SYSTEM

The benchmark system is based on the digital feedwater control system (DFWCS) for an operating pressurized water reactor (PWR). The architecture, systems, and their interconnections of the system have evolved from their analog counterparts to digital ones. However, the system described in Secs. III.A and III.B is used for illustrative purposes only. It has been generalized to be more representative of this class of systems and does not represent a specific plant.

III.A. System Overview

The feedwater system serves two steam generators (SGs) (Fig. 1). Each SG has its own digital feedwater controller. The purpose of the feedwater controller is to maintain the water level inside each of the SGs optimally within ± 2 in. (with respect to some reference point) of the setpoint level (defined at 0 in.). The controller is regarded failed if the water level in an SG rises above $+30$ in. or falls below -24 in. Each digital feedwater controller is connected to a feedwater pump (FP), a main feedwater regulating valve (MFV), and a bypass feedwater regulating valve (BFV). The controller regulates

TABLE I

Desirable Benchmark System Features (Adapted from Ref. 2)

LCC Benchmark System Features[a]	TCC Benchmark System Features[b]
Feature 1. A clock that regulates information sampling from the controlled/monitored process a. regulates measurements, b. may lead to roundoff, c. may lead to truncation. Feature 2. Explicit representation of the power requirements that are needed for the digital systems including a. loss of power, b. low power, c. power spikes. Feature 3. Real-time constraints. Feature 4. A polling capability with a. events occurring in between polls, b. sensors that are being polled failing to report a value. Feature 5. An interrupt capability with a. interrupts occurring simultaneously, b. interrupts occurring at an excessive rate, c. unused interrupts that may be activated. Feature 6. Long-term storage with a. failures that can occur in the retrieval of information, b. failures that can occur in the saving of information, c. LCC requirement 2, d. LCC requirement 3. Feature 7. Computation capability both based on the controlled/monitored process physics and interacting with the process physics a. stimulates interaction with the physical process, b. can produce intermittent and functional failures. Feature 8. A self-diagnostic system where a. contradictory data can be delivered to the system, b. events can occur while in self-diagnostic mode. Feature 9. A watchdog timer with a. instances in which there is no safe state, b. instances in which the watchdog timer fails.	Feature 1. Includes LCC requirements. Feature 2. Networking capability with a. failures in the networked systems, b. failures in connecting components (wires, routers, etc.), c. failures of any protocol used, d. failures as a result of the network topology, e. transient failures in the network. Feature 3. Analog backups to digital systems that include failures in which either the digital or analog system has failed. Feature 4. Shared memory with failures which involve a. data races, b. both deadlocks and starvation. Feature 5. Shared external resources with a. failures involving both deadlocks and starvation, b. network failures. Feature 6. Fault tolerance capability to test Byzantine failures. Feature 7. A database with a. LCC requirement 6, b. failures that can force the database to be inconsistent. Feature 8. Capability to simulate different configurations/versions of software installed on each of the duplicated components and shared resources, including all permutations of homogeneous and heterogeneous software and/or hardware.

[a]LCC systems are those with potential type I coupling between failure events.
[b]TCC systems are those with potential type I and type II coupling between failure events.

the flow of feedwater to the SGs to maintain a constant water level in the SGs. In addition to the FP, the FP seal water system, MFV, and BFV components indicated above, the feedwater control system (FWCS) contains high-pressure feedwater heaters and associate piping and instrumentation.

The FPs are steam turbine–driven, horizontal, double-suction, double-volute, single-stage, centrifugal pumps. The pumps have a design output of 15 000 gal/min at a suction rate of 318.7 psia and a discharge pressure of 1189 psia. The normal operating discharge pressure is ~1100 psig at 100%. The FP is driven by a steam turbine that is dual admission, horizontal, 9140 horsepower, and 5350 revolutions per minute. During plant operation with power >5%, the turbine is aligned to the reheat and main steam system. Steam is supplied from the main steam

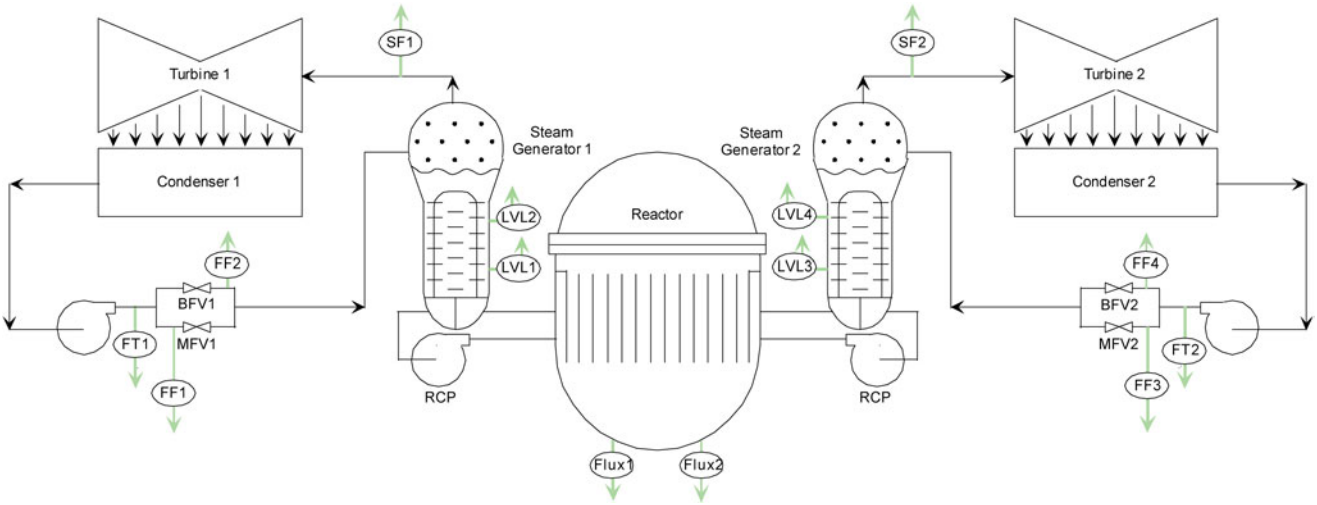

Fig. 1. The benchmark system outlay.

system during plant start-up until the reheat steam pressure is sufficient to supply the turbines. If the main steam is not available or power is $<5\%$, steam can be supplied to the FP turbine from the auxiliary steam system. The purpose of the FPs is to pump the feedwater through the high-pressure feedwater heaters into the SGs with sufficient pressure to overcome both the SG secondary-side pressure and the frictional losses between the FP and the SG inlet. Feedwater regulating valves regulate the amount of feedwater going to the SG in order to maintain a constant water level in the SG.

The MFV is a 10-in., air-operated, angle control valve with 16-in. end connections. The actuator is a piston-type actuator, with separate instrument air supplies to the top and the bottom of the piston. Ball valves control the admission of operating air to the piston for opening and closing operations. The BFV is a 6-in., air-operated, steel control valve.

From an operational point of view, the FWCS operates in different modes, depending on the power generated in the primary system. These modes are the following:

1. low-power automatic mode
2. high-power automatic mode
3. automatic transfer from low- to high-power mode
4. automatic transfer from high- to low-power mode.

The low-power mode of operation occurs when the reactor operates between 2 and 15% reactor power. In this mode, the BFV is used exclusively to control the feedwater flow. The MFV is closed, and the FP is set to a minimal value. The control laws use the feedwater flow, feedwater temperature, feedwater level in the SG, and neutron flux to compute the BFV position. The feedwater level is fed to a proportional-integral controller using the feedwater temperature to determine the gain. Then, this gain value is summed with the feedwater flow and neutron flux. Essentially, neutron flux and feedwater flow are used to predict the change in water level (see Sec. III.B).

The high-power mode is used when the reactor power is between 15 and 100% reactor power. In this mode, the MFV and the FP are used to control the feedwater flow. The BFV is closed in a manner that is similar to the low-power mode. The control laws (see Sec. III.B and Appendix A) use the feedwater level in the SG, steam flow, and feedwater flow to compute the total feedwater demand. The feedwater flow and steam flow are summed and fed to a set of proportional integral controllers. The output from these controllers is added to the feedwater level, and that result is fed to a proportional integral controller that uses the steam flow for the controller's gain. The total feedwater demand is used to determine both the position of the MFV and the speed of the FP. The FP also uses the other digital feedwater controller's MFV output to compute the speed needed.

Each digital feedwater controller comprises several components (Fig. 2), which provide both control and fault-tolerant capabilities. The control algorithms (see Secs. III.B, III.C, and Appendix A) are executed on both a main computer (MC) and backup computer (BC). These computers produce output signals for the MFV, BFV, and FP. The selection of the appropriate signal to be used (from the MC or BC) is determined by a controller for each of the respective actuated devices (i.e., MFV, BFV, and FP). Each of these controllers can forward the MC or BC outputs to the respective actuated device, or it can maintain the previous output to that device. If the controllers decide to maintain a previous output value to a controlled device, it is necessary for operators to override the controller (Sec. III).

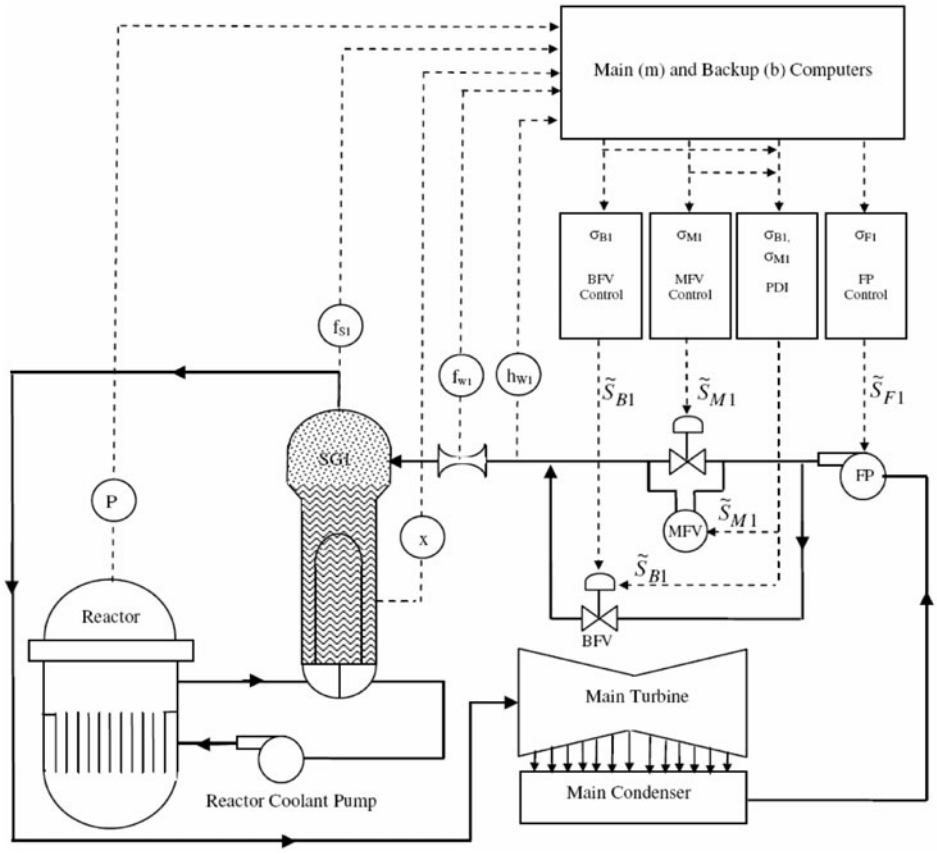

Fig. 2. Detailed view of a single feedwater controller. Solid lines indicate piping. Dashed lines indicate signals.

Transitions between low and high power are controlled by the neutron flux readings. When the system is in the low-power mode and the neutron flux increases to a point when the high-power mode is necessary, the MFV is signaled to open while the BFV closes to maintain the needed feedwater flow. The analogous situation occurs when the system is in the high-power mode and the neutron flux decreases to a point when the low-power mode is needed.

III.B. Detailed View of the Benchmark System

This section describes the digital feedwater controller at a greater level of detail. In particular, the physical connections between the sensors, computers, and valve actuators are examined (Sec. III.B.1), and a comparison of the benchmark system with the features listed in Sec. II and in Ref. 3 is presented (Sec. III.B.2). More detailed information concerning the benchmark is provided in Appendixes A through D. More specifically, the detailed control laws applied in the digital I&C system, the fault-tolerant features of the architecture, and the system failure modes are described in Appendixes A, B, and C, respectively. A discrete state representation of the benchmark DFWCS is also provided in Appendix D.

III.B.1. Physical Connections

The DFWCS obtains information about the state of the feedwater system through the use of several sensors that measure feedwater level, neutron flux, feedwater flow, steam flow, and feedwater temperature (Fig. 2). As shown in Figs. 3 through 7, the sensor signals are routed to provide information to both the MC and BC. Setpoint data are delivered from the MFV controller to the MC and BC through an analog signal.

The DFWCS components are connected together in several different ways as shown in Figs. 8 and 9. First, both the MC and BC connect to the MFV, BFV, and FP controllers through an analog control signal and failure status signals. The MC and BC are required to respond within 750 ms upon receiving a signal. The MFV, BFV, and FP controllers are connected so they may share status information. Another controller, the pressure differential indicator (PDI) controller, serves as a backup for the MFV controller. This PDI controller reads the

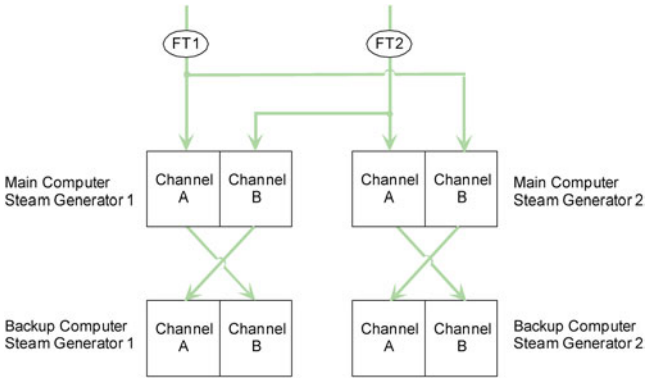

Fig. 3. Feedwater temperature sensor signals.

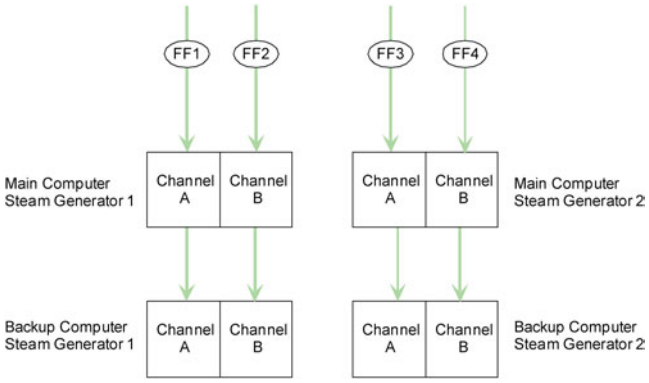

Fig. 4. Feedwater flow sensor signals.

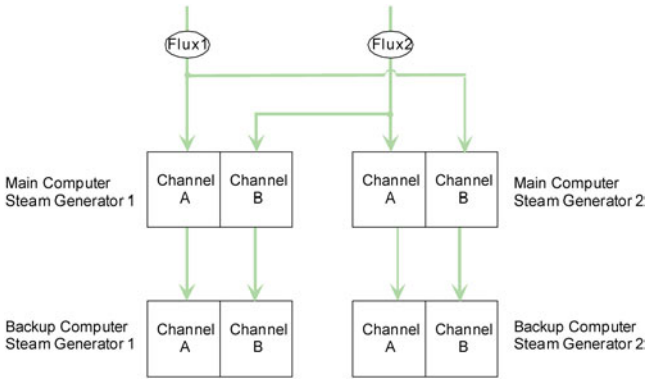

Fig. 5. Neutron flux sensor signals.

value of the signal originating from the MFV controller. If the MFV controller fails to send a signal, the PDI will produce the most recent value of the signal on the MFV controller's output to the MFV. The PDI is also connected to the BFV, MFV, and FP controllers in the

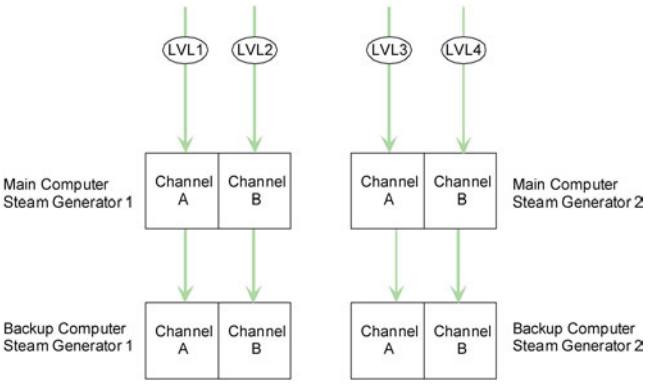

Fig. 6. Feedwater level sensor signals.

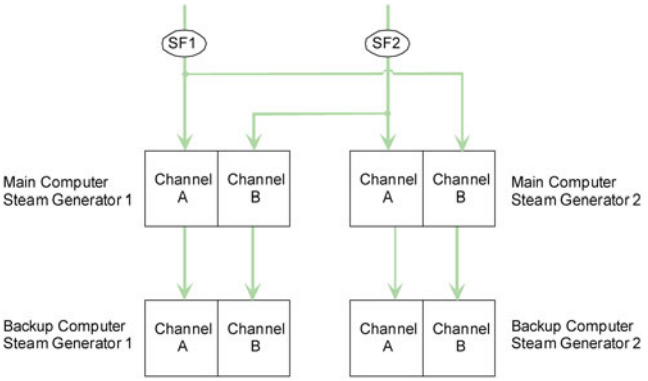

Fig. 7. Steam flow sensor signals.

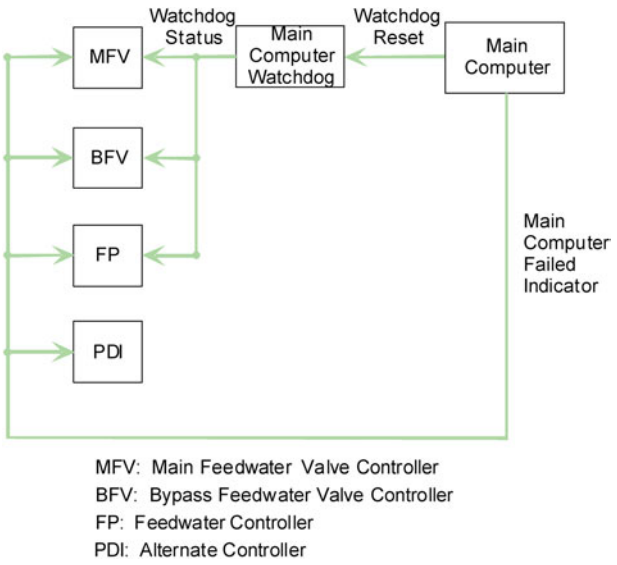

Fig. 8. Digital feedwater controller status interconnections for MC.

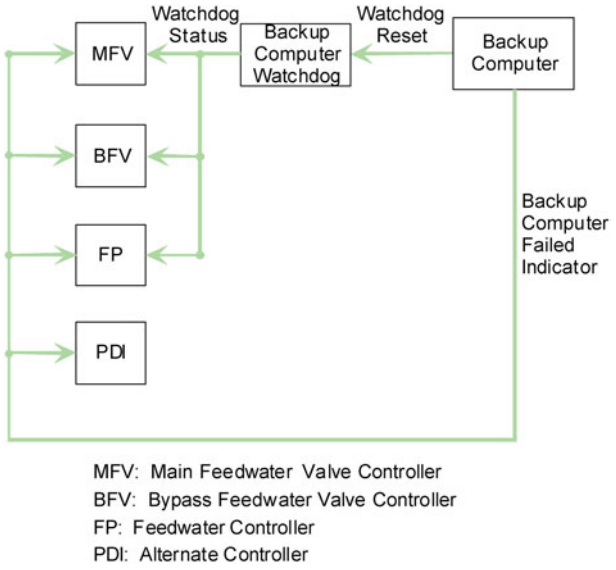

Fig. 9. Digital feedwater controller status interconnections for BC.

same manner as those controllers are connected to each other.

III.B.2. Comparison of Benchmark System with Desirable Features from Ref. 3

In this section, we briefly discuss the features of the presented benchmark system with those desirable features presented in Ref. 3 and reiterated in Table I. Such a comparison was performed in Ref. 25, which presents several scenarios based upon Licensee Event Reports to demonstrate the ability of the benchmark system to meet LCC features 2, 4, 7, 8 and 9. These features, respectively, include the representation of power, device polling, interaction with the plant process, a self-diagnostic system, and the use of watchdog timers. However, this benchmark system also incorporates the other LCC desirable features by the

1. use of computers for the control of the system (LCC feature 1)
2. requirement for the action of the computers within 750 ms (LCC feature 3)
3. capability to use of changeable setpoints and the control tunables that can be stored by the digital system (LCC feature 6)
4. processing of interrupts via the watchdog timer[d] (LCC feature 5).

[d] When a watchdog timer goes off, it signals a device (here the device would be one of the MFV, BFV, or FP controllers) that uses interrupts.

The benchmark system, however, does not incorporate most of the TCC features in Table I. For example, it does not include a database. This is expected, as the desirable TCC features, as discussed in Ref. 3, are designed to include more complicated systems than currently in use while LCC features are designed to be applicable to digital I&C systems that are currently in use.

III.C. An Example Initiating Event for Illustration

The following initiating event is used to illustrate how a failure scenario, to be investigated as part of a PRA and/or reliability analysis, can be defined for the benchmark system and investigated with the DFM and Markov/CCMT modeling and analytical approaches (Sec. IV):

Assumption 1. Reactor is shut down and power P is generated from the decay heat.

Assumption 2. Reactor power output drops to 6.6% of 3000 MW(thermal) [or 1500 MW(thermal)/SG] 1 s after reactor shutdown, with steam flow from the SGs following according to the overall plant system time lag and control characteristics.

Assumption 3. Feedwater flow is at nominal level.

Assumption 4. Off-site power is available.

Assumption 5. MC is failed.

Assumptions 1 through 4 are consistent with the events following a turbine trip. Assumption 5 is made to reduce the state-space for clarity in illustrating the DFM and Markov/CCMT model construction.

Since the plant is in post-reactor-shutdown (low-power) mode, the BFV is being utilized (see Sec. III.A). Then, from Appendix A, the dynamic behavior of the system with which both the DFM and Markov/CCMT models have to be consistent can be represented via Eqs. (A.1) through (A.13) while also reflecting the assumptions 1 through 5 as the specific boundary conditions and constraints governing the system behavior for this scenario.

Figures 10, 11, and 12 show the behavior of the system for the reference conditions assumed for this illustration. Both level x_n and compensated level C_{Ln} (see Nomenclature on p. 93) stabilize around their nominal value within 100 s following the initiating event, while level error E_{Ln} shows a steady decrease after 100 s. This behavior is consistent with the definition of $E_{Ln}(t)$ as the difference between the setpoint r_n and $C_{Ln}(t)$. The compensated level $C_{Ln}(t)$ anticipates the behavior of the difference between the steam outflow and the feedwater inflow into the SG. Since the steam outflow follows the power generated in the primary system and power decreases with time, so does the difference between the actual level x_n and anticipated level $C_{Ln}(0)$.

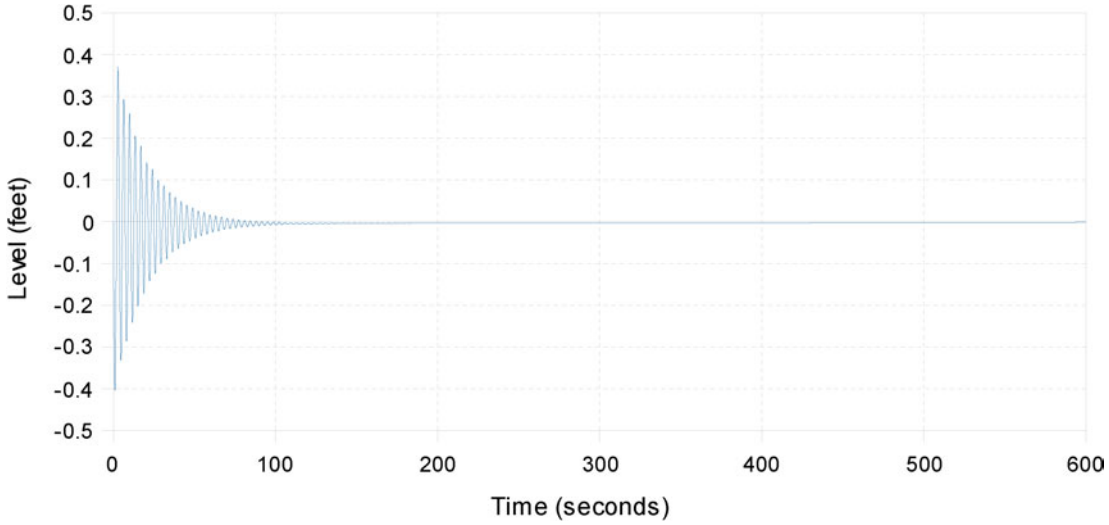

Fig. 10. Variation of level with time for the example initiating event.

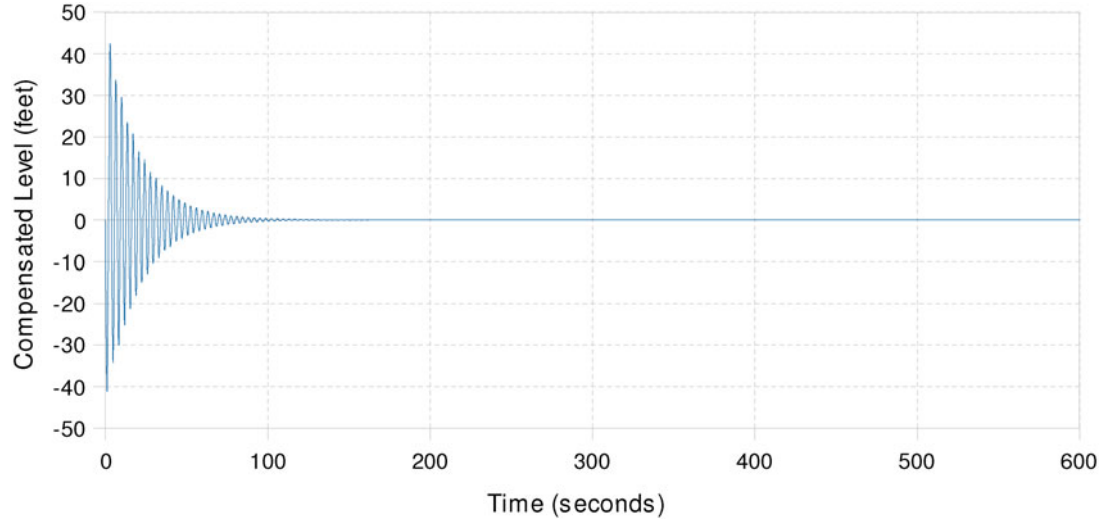

Fig. 11. Variation of compensated level with time for the example initiating event.

Figure 13 shows that the exact timing of the failure of a system component can have an impact on the resulting system failure. In particular, Fig. 13 depicts the evolution of the level variable under two distinct scenarios starting both from the same initial conditions as those in Fig. 10. In one case, the BFV fails stuck at the current position at time $t = 43$ s. In the other case, the BFV fails stuck at time $t = 44$ s. The first scenario results in the level failing low ($x_n < -2.0$ ft), while the second scenario results in the level failing high ($x_n > 2.5$ ft). This example is important because for a system similar to the DFWCS in an operating PWR, it illustrates (a) what has been reported in the literature on the possible sensitivity system failure mode to the exact timing of component failures[26] and (b) that an analysis that considers only the order of events and ignores their exact timing may result in the failure to identify the correct failure mode, which may or may not be risk significant.

Figures 14, 15, and 16 present another interesting issue. Figures 14, 15, and 16 display the same data shown in Figs. 10, 11, and 12 except that they include a longer time interval ($0 \leq t \leq 1200$ s). The system seems to exhibit instability around time $t = 880$ s where the three variables start oscillating again. The level and the compensated level quickly settle again around their nominal value, and the level error seems to make a jump before resuming its slow descent. This behavior may be caused by an actual instability in the system and its

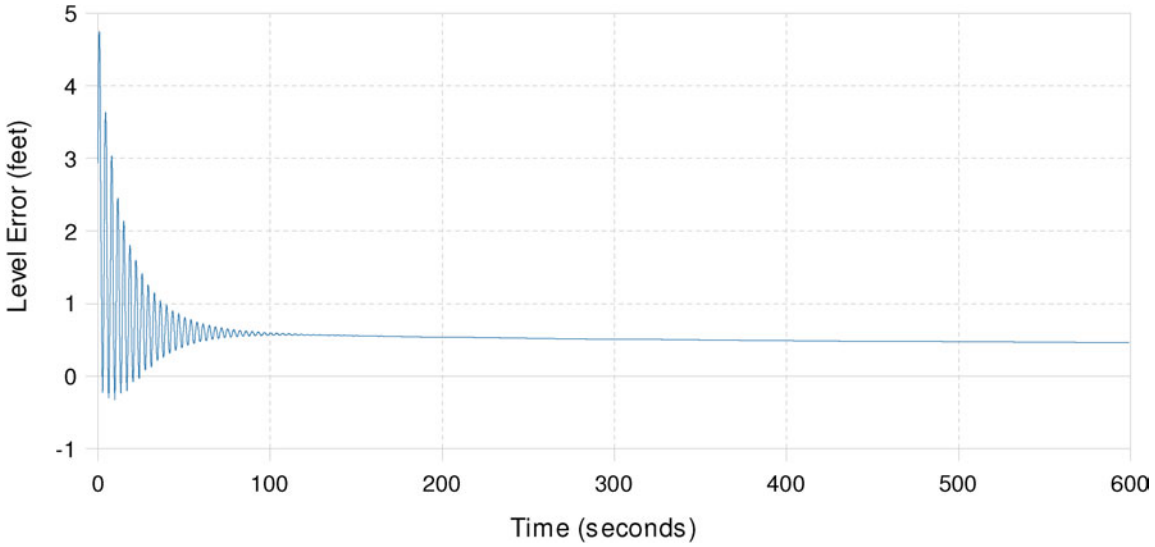

Fig. 12. Variation of level error with time for the example initiating event.

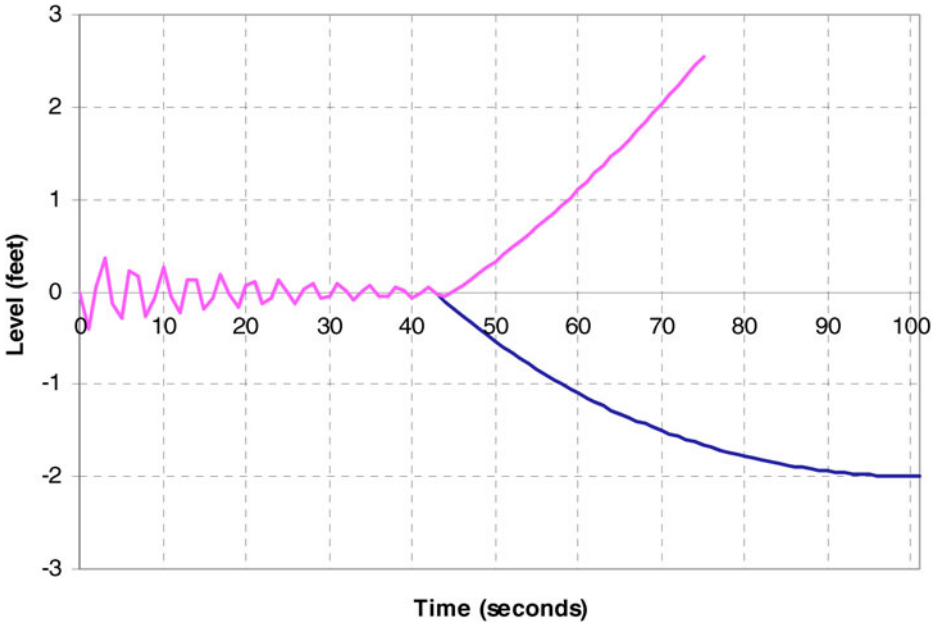

Fig. 13. Different failure modes as result of timing of BFV failure.

corresponding model. Such instabilities have been observed in nuclear plants.[e] However, in this case, it is an artifact that is the result of a numerical error in the digital control algorithm simulator. The algorithm uses Gauss-Legendre quadrature to evaluate the integral in Eq. (A.33) in Appendix A. The integral is computed repeatedly with an increasing number of points until the absolute value of the difference between two consecutive estimates of the integral is below a given threshold (10^{-6}). At time $t = 880$ s, the first two estimates of the integral are both below the threshold itself, so that the absolute value of their difference is also below the

[e]Neutron flux oscillations with scram following recirculation pump trip were observed in La Salle Unit 2, Illinois, on March 9, 1988; power oscillations after a turbine trip with pump runback were observed in Oskarshamn Unit 2, Sweden, on February 25, 1999; feedwater oscillations were observed in Harris plant, North Carolina, during start-up at 7% power on January 2, 2002. Also see Ref. 27.

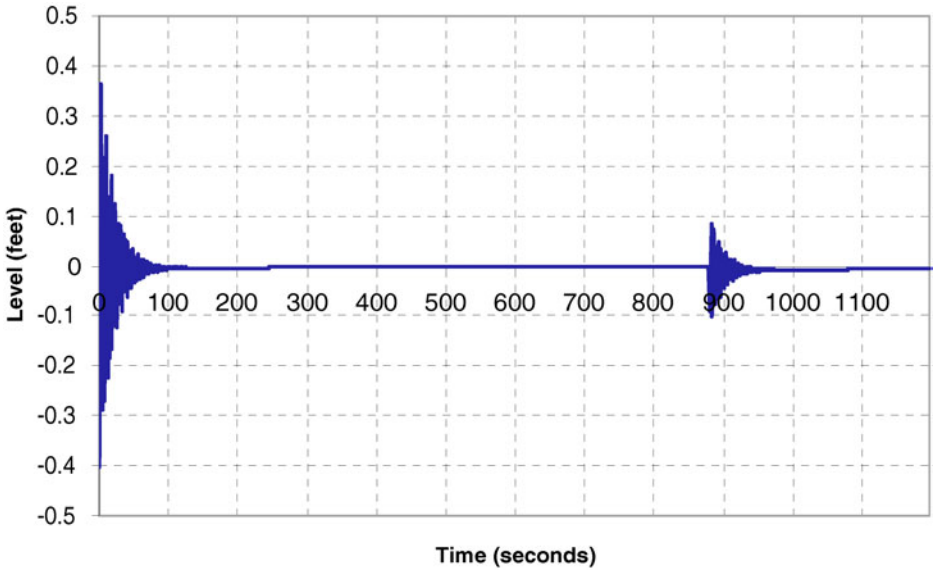

Fig. 14. Variation of level with time with artifact.

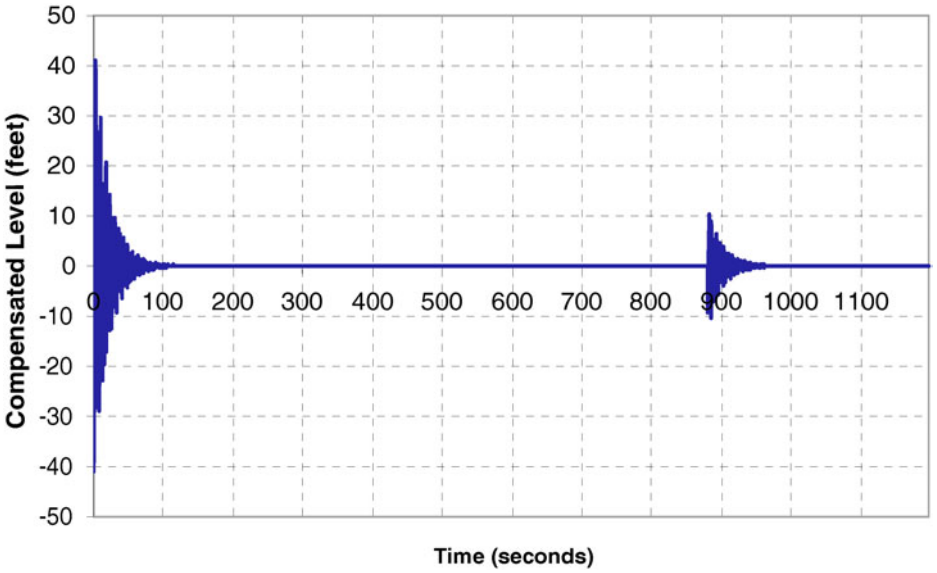

Fig. 15. Variation of level error with time with artifact.

threshold. This causes the algorithm to stop its iteration and return the wrong value for the integral. Figures 17 and 18 show the correct integral in the range $0 \leq t \leq 1200$ s, and the integral calculated by the faulty algorithm in the same time interval, respectively. The numerical problem presented here would probably be avoided by an experienced, qualified programmer. However, this example is important because it illustrates the kind of pitfalls that can arise in the presence of digital systems and software control algorithms.

IV. OBTAINING THE PRIME IMPLICANTS FOR SYSTEM TOP EVENTS

In general terms, the failure and reliability analysis of a digital I&C system can follow the same logical steps of the FT analysis of a conventional hardware system. More specifically, in the context of a nuclear power plant PRA, FT top events are defined as corresponding "initiating events" or "pivotal events" in ET sequences corresponding to specific risk scenarios. Thus, when such

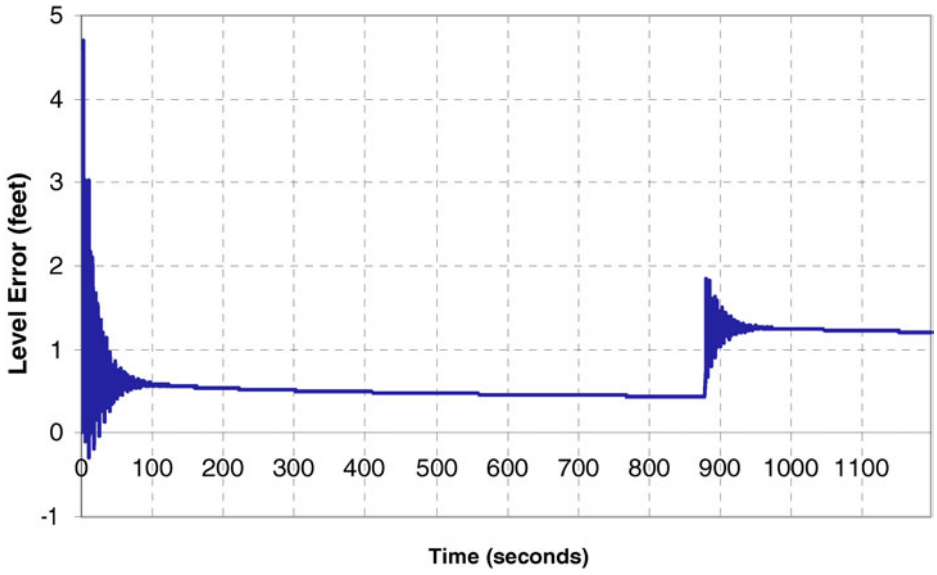

Fig. 16. Variation of compensated level with time with artifact.

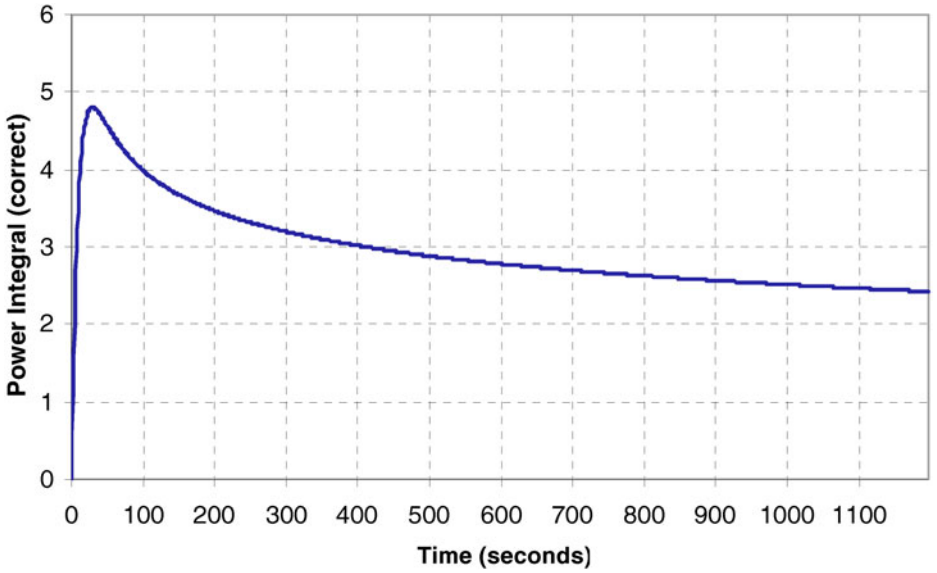

Fig. 17. Correct evaluation of the integral in Eq. (33).

events correspond to digital I&C related events for a system for which a DFM and/or Markov/CCMT model has been constructed, the system failure analysis process can proceed as follows:

1. Define a top event (or several top events) that define the system failure(s) of interest—for example, in our case, such a top event definition could be "SG level fails high or low"—and translate the nominal top event definition into the equivalent logic statements that apply in the context of the particular type of model (e.g., DFM or Markov/CCMT) being used for the analysis.

2. Utilize the system model constructed in the particular paradigm chosen (DFM and/or Markov/CCMT) to identify prime implicants for the top event(s) of interest.

3. Quantify the prime implicants obtained to obtain estimates of the top event failure probability (or frequency) and therefore of the I&C system reliability.

Some observations are in order to further clarify the above. Mention has already been made of the similarity and differences between coherent FT cut sets and multivalued logic/noncoherent binary logic prime implicants.

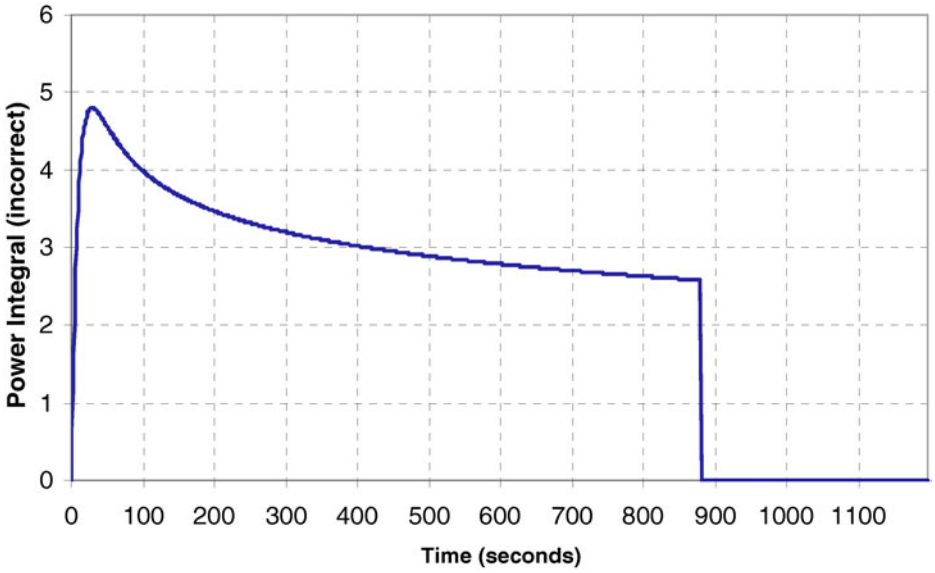

Fig. 18. Incorrect evaluation of the integral in Eq. (33).

A more important difference between FT analysis and DFM or Markov/CCMT analysis is that while an FT is a model that is specific to a particular type of system failure event (i.e., the one defined by the top event itself) and is developed ad hoc for that event, a DFM or Markov/CCMT model is a full functional model of the entire system of interest, *which can be analyzed for any number of different top events*. This feature is often overlooked by practitioners who cite the complexity of the latter type of techniques as an impediment to their use, but it needs to be taken into due consideration. Although it is true that building a good system DFM or Markov/CCMT model is not trivial (it cannot be, given the complexity of the systems for which this type of modeling is needed), it is also true that once built, the model can be reused for an unlimited number of distinct analyses and to obtain prime implicants for a broad variety of separate top events. Thus, the same DFWCS model can be analyzed in the context of a variety of risk scenarios, such as for determining the causes and probabilities for high SG level while the system is in automatic mode, but also for when it is in manual mode or in turbine bypass mode. The same is true for the low SG level top event or for a loss-of-feedwater top event or for a loss-of-bypass-control capability or for any other type of system failure that may be of interest as the initiating event or the pivotal event of a plant risk scenario.

Another observation, which perhaps represents one of the principal lessons learned from the modeling and analytical activities carried out in the study summarized in this paper, concerns the complementary nature of the two analytical methods applied. As will be further explained in Secs. V and VI, both DFM and Markov/CCMT model a system by discretizing its states, including those represented by continuous variables, into a finite set. Subsequently, both types of models correspond to a considerably more detailed and accurate system representation than what is found in a traditional binary model. DFM, however, is typically and more naturally applied in a coarser mode of modeling than Markov/CCMT. Degree of coarseness in discretization is actually what in the end determines whether a model can be analyzed exhaustively in deductive mode (i.e., from effect to causes, like an FT). Although both DFM and Markov/CCMT can operate both in the deductive and inductive modes (i.e., starting from assumed initial conditions and marching forward in causality and time flow), normally DFM uses a relatively simplified representation of a system, which can then be completely and exhaustively investigated by deductive analysis without running against the limits of current computer processor and memory capabilities. On the other hand, Markov/CCMT may generally provide a more detailed representation and detail of analytical results, but these results can, however, be obtained only inductively or by considering all transitions between system states (see Sec. VI). The inductive approach raises questions of completeness, and the consideration of all the transitions between system states may not be practical for large systems because of the computational requirements.

From the above observations a firm indication is that the most effective way to apply the methodologies considered in our study is in a complementary fashion, by which (a) the deductive analysis of DFM is used first to carry out in multivalued logic coarse mode the formal identification of the full spanning of potential

risk scenarios and (b) then the more detailed temporal representation of Markov/CCMT is applied to further investigate and assess risk-relevant variations of each coarse class of system failure modes and scenarios identified by DFM to make sure that the coarseness of DFM has not masked out any important variations of the failure modes identified.

It is worthwhile noting that the foregoing recommendation concerning the mixed use of deductive and inductive analyses has general validity and has been routinely applied in the common PRA practice. For example, deductively defined master logic diagrams[28] are used in PRA to identify scenario initiating events that are then explored and developed via ETs. Then again, at the system or subsystem PRA modeling level, deductive FT analyses for specific system top events are complemented by inductive failure mode and effect analyses (FMEAs) to validate the accuracy of the FT cut sets that are identified.

Sections V and VI, respectively, discuss the complementary DFM and Markov/CCMT analyses that were executed.

V. SYSTEM FAILURE ANALYSIS USING DFM

This section discusses the application of DFM to the benchmark system example initiating event presented in Sec. III.C. A brief overview of the methodology is given below, before proceeding to the specific discussion of the DFM model that was constructed to represent the benchmark system (Sec. V.A), of the analyses that were executed, and of the top event prime implicants that were identified on the basis of such a model and analyses (Sec. V.B).

The DFM is a methodology for system analysis that has been demonstrated in several NRC and National Aeronautics and Space Administration applications over the past 10 yr (Refs. 13 and 29 through 33). It combines multivalued logic modeling and analysis capabilities and can be integrated with an ET/FT PRA logic structure in relatively straightforward fashion. In practical terms, DFM is implemented in the software toolset DYMONDA™, which permits the construction and editing of DFM system models, as well as their analysis via automated deductive and inductive formal logic functions and algorithms that a user can select and apply.

DFM has several unique features that address digital systems:

1. the capability to model and analyze feedback loops and time transitions
2. deductive and inductive modules that can analyze detailed multivalued logic models to identify and characterize interactive failure modes and software error forcing contexts. The deductive module explores the causality of the system model in reverse and generates prime implicants that can be thought of as a multivalued logic equivalent of minimal cut sets. The inductive module follows the causality of the system model and produces automated system FMEA trees, to verify expected behavior.
3. the capability to quantify the top events analyzed by the deductive analysis module, in a fashion compatible and easily integrated with standard PRA quantification processes.

In applying DFM to the benchmark system, a system model encompassing both the digital controllers and the process being controlled (i.e., the SG and the feedwater system) was constructed. This model can be used in conjunction with the plant ET/FT PRA models. More specifically, some of the plant PRA ETs contain pivotal events that are tied to the failure of the FWCS, which in our case is assumed to be the benchmark DFWCS. Thus, instead of expanding these pivotal events with FT models, the DFM model of the benchmark system is analyzed and solved. The prime implicants and/or probability estimates obtained with DFM analyses can then be exported back into and integrated with the plant PRA models.

The essential steps in applying DFM in a PRA framework are the following:

1. Construct a DFM model to represent the system of interest.
2. Analyze the DFM model.
3. Quantify the results.

These three essential steps are covered below with specific reference to their application to the analysis of the benchmark DFWCS system.

V.A. Benchmark System DFM Model Construction

A DFM model is a graphic network that links key process parameters to represent the cause-and-effect and the time-dependent relationships for a system of interest. In particular, for a digital control system, both the controlled/monitored process and the controlling software itself are represented in the DFM model.

Key controlled/monitored process parameters and software variables that capture the essential behavior of these components and software/firmware functions are identified and represented as process variable nodes. These process variable nodes are then linked together through transfer boxes or transition boxes for instantaneous actions or time-delayed actions, respectively. Detailed transfer functions that model the relationships between these parameters are represented as decision tables, which in essence are the multilogic extension of binary truth tables. Discrete behaviors such as component failures and logic switching actions are identified and represented in DFM

as condition nodes, which act as switches that "activate" in the DFM model the portion of internode transfer functions that represent the specific cause-effect and temporal relationship between process variable nodes that governs such variable at a particular time and under particular overall system circumstances.

The DFM decision tables can be constructed by empirical knowledge of the system, from the equations that govern the system behavior, and from the available software code and/or pseudo code. In particular, when modeling a system that includes actual software, software module and unit testing (which itself constitutes the basic first step of standard software testing procedures) becomes an integral part in the creation of the decision tables that mimic the actual behavior of the software.

The DFM model developed to analyze the benchmark system example initiating event is shown in Fig. 19. This model encompasses the BC, the BFV, the BFV controller, the inputs and outputs for the BC, and the control law and logic for maintaining the SG level. Thus, the hardware, the software, and their interactions are all included in one system model, and in this model the process variable nodes are each discretized into a finite number of states. For example, the discretization of node BFV (the bypass flow valve condition, a hardware variable) is shown in Table II and reflects the failure modes assumed for that particular component. As a further example, the discretization of EL (the internal software variable representing the SG level error) is shown in Table III and reflects the possible range of values for that specific software variable.

TABLE II

Discretization of Node BFV

State	Description
OK	Bypass flow valve is OK.
F-S	Bypass flow valve failed stuck.

TABLE III

Discretization of Node EL

State	Description
−1	[−1000, −200)
0	{−200, 200)
+1	[200, 1000]

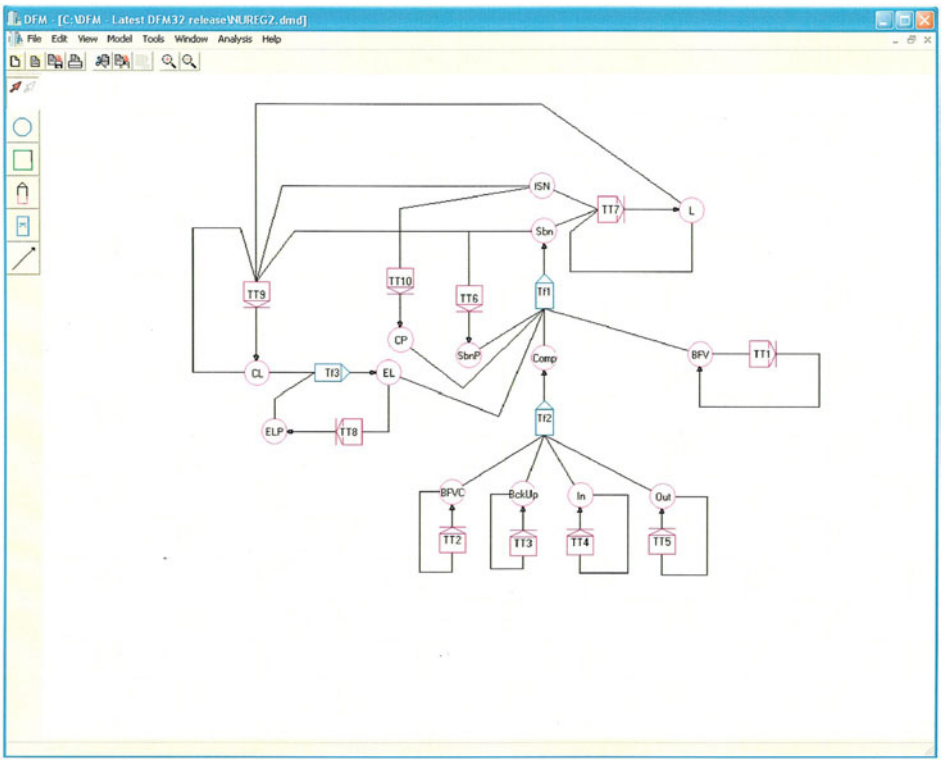

Fig. 19. DFM model of the benchmark system initiating event.

The process variable nodes are linked together in the DFM system model to represent the temporal and causal behavior of the system, in general terms but of course also more specifically for the circumstances corresponding to the example initiating event condition that is of interest here. For example, transfer box Tf2 in the bottom center portion of Fig. 19 shows that with the MC out of commission, the computer system depends on the BC, on the inputs and outputs to the BC, and on the BFV controller. The relationships between the nodes are summarized in the decision tables. The decision tables for transition boxes TT6, TT7, TT8, TT9, and TT10 and transfer boxes Tf1 and Tf3 are developed from the control equations implemented in the software controller. The decision tables for the other transfer boxes and transition boxes reflect the known logic behavior of the system.

Tables IV and V show examples of the decision tables developed for this model. Table IV is a decision table for transition box TT7. It shows how the current BFV position [node Sbn (DFM node for the BFV position)], the current steam flow [node fSN (DFM node for the steam flow)], and the current SG level [node L (DFM node for the SG level)] will influence the SG level (node L) in the next time step. Thus, the decision table for this transition box models the dynamic behavior of a portion of the system. Table V on the other hand is a decision table for transfer box Tf2. It determines the state of the computer system [Comp (DFM node for the BC)] based on the states of the BFV controller [node BFVC (DFM node for the BFV controller)], the BC (node BckUp), the inputs to the BC (node in), and the outputs of the BC (node out). Specifically, this transfer box indicates among other things that the computer system will fail if the BFV controller fails, if the BC is down, if the inputs are lost, or if the outputs are lost. No time-dependent ("dynamic") information is included in this decision table.

TABLE IV

Decision Table for Transition Box TT7

Sbn	fSN	L	L
0	0	−2	−2
0	0	−1	−1
0	0	0	0
0	0	+1	+1
0	0	+2	+2
0	1	−2	−2
0	1	−1	−2
0	1	0	−1
0	1	+1	0
0	1	+2	+1
:	:	:	:

TABLE V

Decision Table for Transfer Box Tf2

BFVC	BckUp	In	Out	Comp
OK	OK	OK	OK	OK
Failed	—	—	—	Failed
—	Down	—	—	Failed
—	—	Loss	—	Failed
—	—	—	Loss	Failed

V.B. Benchmark System DFM Model Analysis

The analysis of a DFM system model can be conducted by tracing sequences of events either backward from effects to causes (i.e., deductively) or forward from causes to effects (i.e., inductively) through the model structure.

The deductive engine backtracks the time and causality of the DFM model to identify timed prime implicants[29,34] (TPI) for top events of interest. These TPI, characterized by the combinations and sequences of basic variable states, represent the formally complete set of minimal conditions that would lead to the top event. Prime implicant completeness is guaranteed by the use of appropriate logic theorems and formalism in the DFM DYMONDA™ deductive engine algorithms.[29] In this context, "completeness" means that all combinations (exclusive of course of nonminimal combinations) of system parameter and variable states that are implicitly or explicitly included in the original model and that are relevant as root causes of the top event from which the DFM deductive search proceeds are identified. That is, in logic terms, prime implicants are the multivalued logic equivalent of minimal cut sets in traditional FT analysis. The DFM prime implicants are logically compatible with cut sets produced by PRA tools such as SAPHIRE (Ref. 35), CAFTA (Ref. 36), or RISKMAN (Ref. 37). Hence, DFM results can also be exported into a PRA tool environment with a minimum amount of formatting and reformulation.

In a DFM deductive analysis, dynamic consistency rules may be used to prune out conditions that are not compatible with the dynamic constraints of the system of interest. This generally makes the analysis more efficient as well as more accurate. For instance, dynamic consistency rules can be defined to constrain (a) the direction of change of certain parameters—for example, if repair is not available, a component, once enters into a failed state, remains in that state—or (b) the rate of change of certain parameters.

Besides the deductive engine, the inductive engine can be executed to determine how a particular set of basic variable states (the initial condition) produces various

sequences and system-level states. Starting from a set of initial conditions, the inductive engine follows the causality and timing represented in the model to determine the resulting sequence of events.

Via its deductive and inductive analytical modes, DFM provides the multistate and time-dependent equivalent of both ET/FT analysis and FMEA. As mentioned earlier, an advantageous feature of a DFM system model is that once the model has been developed, it can repeatedly be analyzed by automated execution, deductively and/or inductively, for any variety of top events and scenario sequences that are believed to be risk relevant. This is more efficient compared to the manual development and integration of individual ET and FT models for each event or sequence that needs to be carried out with the classical ET/FT PRA techniques.

Another useful characteristic of DFM models is that they represent both the success and failure sides of system behavior and functionality. Thus, DFM inductive and deductive analyses can be combined to analyze a system not only within the context of fault analysis but also for the purpose of design validation and verification and automated test sequence generation. A discussion of DFM usage for all three of these system analysis objectives can be found in Refs. 29, 31, 32, and 33. For the DFWCS benchmark system, we limit the following discussion to presenting the deductive and inductive analyses that were executed for fault and failure-mode identification purposes, in relation to the specific initiating event that was defined in Sec. III.C.

In the mutually complementing, combined usage that we recommended to obtain maximum benefit from the DFM and Markov/CCMT techniques, a DFM deductive analysis would be the first step, to yield the initial identification of a logically self-consistent and "complete" set of system failure modes and root conditions, expressed in the form of prime implicants. Inductive analyses would then be performed using the more detailed modeling capabilities of Markov/CCMT (Sec. VI), especially in terms of timing and fault-recovery effects. DFM inductive analyses can also be carried out in parallel to the Markov/CCMT analyses, as a further form of model validation.

Two DFM deductive analysis examples are provided in Sec. V.B.1, and two inductive analysis examples are presented in Sec. V.B.2.

V.B.1. Deductive Analysis of the Benchmark System

For the example initiating event, failure and fault analyses using the deductive technique were carried out to find out the combination of component states that could lead to desirable or undesirable events of the DFWCS.

For the failure and fault analysis example, to find out the prime implicants for a high SG level, the following top event was defined as follows:

$$\begin{aligned} \mathrm{L} &= +2 \quad \text{at } t = 0 \quad \text{AND} \\ \mathrm{L} &= +1 \quad \text{at } t = -1 \quad \text{AND} \\ \mathrm{L} &= 0 \quad \text{at } t = -2 \quad \text{AND} \\ \mathrm{ELP} &= 0 \quad \text{at } t = -2 \quad \text{AND} \\ \mathrm{CL} &= 0 \quad \text{at } t = -2 \ , \end{aligned}$$

where ELP is the DFM node for the previous level error and CL is the DFM node for the compensated level.

This top event specified the progression of the SG level from 0 to 2, given nominal values of the level error and compensated level in the control software. In the deductive analysis of this top event, the top event can be expressed as a transition table, as shown in Table VI. The header row shows the nodes and their associated time stamp, and row 1 shows the combination of the states for the nodes of interest.

In this deductive analysis, the model was tracked backward in time and causality, as explained for illustration below. With the analysis time set to 0, the DFM deductive engine uses the decision table for transition box TT7 to expand the top event definition given by Table VI. In particular, this expansion identifies the combinations of fSN, Sbn, and L states at $t = -1$ that give rise to L = 2 at $t = 0$. The result of the expansion was the transition table shown in Table VII.

TABLE VI

Transition Table for the Top Event

L $t = 0$	L $t = -1$	L $t = -2$	ELP $t = -2$	CL $t = -2$
+2	+1	0	0	0

TABLE VII

Transition Table for After the First Expansion

Sbn $t = -1$	fSN $t = -1$	LP $t = 0$	L $t = -1$	L $t = -2$	ELP $t = -2$	CL $t = -2$
—	0	+2	+1	0	0	0
1	0	+1	+1	0	0	0
2	0	+1	+1	0	0	0
1	1	+2	+1	0	0	0
2	1	+1	+1	0	0	0
2	—	+2	+1	0	0	0

To continue the deductive analysis, the causality shown in the model is further backtracked by the deductive algorithm. For the transition table shown in Table VII, the first column, corresponding to Sbn at $t = -1$, is next expanded with the decision table for transfer box Tf1. This process is repeated, with appropriate logic reductions and constraint enforcement[29] until the whole model is traversed backward for two time steps. The prime implicants shown in Table VIII are the product of this process. In formal logic terms, these prime implicants describe the combinations of basic events that could cause the top event, but none of these prime implicants is contained in another; i.e., these prime implicants are in essence the multivalued logic equivalent of minimal cut sets in an FT analysis:

$$\textit{Top Event} = \textit{Prime Implicant}\ \#1 \vee \cdots \vee \textit{Prime Implicant}\ \#10\ ,$$

$$\textit{and Prime Implicant}\ \#I \subseteq \textit{Prime Implicant}\ \#j\ .$$

For the top event of interest, prime implicants #1 through #4 and prime implicants #6 through #9 identified the conditions that the BFV failed stuck, loss of inputs of the computer, the downing of the computer, or the freezing of the BFV controller, together with a steam flow-feed flow mismatch (feed flow > steam flow) will cause the SG level to rise. This is because any of the failure will cause the feed flow to remain the same, while the steam flow gradually decreases. On the other hand, prime implicants #5 and #10 identified the condition corresponding to the BFV failure in an arbitrary state.

If the probabilities for the basic event nodes (those that are not downstream of transfer boxes) in Fig. 19 are defined, the top event can be quantified using the procedure outlined in Sec. V.B.3; that is, the set of prime implicants is first converted into a set of mutually exclusive implicants, so that the sum of the probabilities of these mutually exclusive implicants yields the probability of the top event:

TABLE VIII

Prime Implicants for High SG Level

Number	Prime Implicant	Number	Prime Implicant
1	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 1 at $t = -2$ fSN = 0 at $t = -2$ BFV = F-S at $t = -2$	6	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 2 at $t = -2$ fSN = 1 at $t = -2$ BFV = F-S at $t = -2$
2	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 1 at $t = -2$ fSN = 0 at $t = -2$ Comp = LossIn at $t = -2$	7	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 2 at $t = -2$ fSN = 1 at $t = -2$ Comp = LossIn at $t = -2$
3	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 1 at $t = -2$ fSN = 0 at $t = -2$ Comp = Down at $t = -2$	8	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 2 at $t = -2$ fSN = 1 at $t = -2$ Comp = Down at $t = -2$
4	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 1 at $t = -2$ fSN = 0 at $t = -2$ BFV = Frz at $t = -2$	9	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 2 at $t = -2$ fSN = 1 at $t = -2$ BFV = Frz at $t = -2$
5	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 1 at $t = -2$ fSN = 0 at $t = -2$ BFV = Arb at $t = -2$	10	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 2 at $t = -2$ fSN = 1 at $t = -2$ BFV = Arb at $t = -2$

$$Top\ Event = \text{MEI \#1} \vee \cdots \vee \text{MEI \#}m \ ,$$

where

$$\text{MEI \#}I \wedge \text{MEI \#}j = \varnothing$$

$$P(Top\ Event) = P(\text{MEI \#1}) + \cdots + P(\text{MEI \#}m)$$

and MEI is the mutually conclusive implicant.

As previously mentioned, once a single DFM model is constructed, it can be analyzed for many different top events. For example, the same DFM model could be analyzed for the top event:

$$\begin{aligned} \text{L} = -2 \quad & \text{at } t = 0 \quad \text{AND} \\ \text{L} = -1 \quad & \text{at } t = -1 \quad \text{AND} \\ \text{L} = 0 \quad & \text{at } t = -2 \quad \text{AND} \\ \text{ELP} = 0 \quad & \text{at } t = -2 \quad \text{AND} \\ \text{CL} = 0 \quad & \text{at } t = -2 \ . \end{aligned}$$

This top event specified the progression of the SG level decreasing from 0 to −2, given nominal values of the level error and compensated level in the control software. For this particular top event, the 11 prime

TABLE IX

Prime Implicants for Low SG Level

Number	Prime Implicant	Number	Prime Implicant
1	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 0 at $t = -2$ fSN = 1 at $t = -2$ BFV = F-S at $t = -2$	7	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 1 at $t = -2$ fSN = 2 at $t = -2$ BFV = F-S at $t = -2$
2	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 0 at $t = -2$ fSN = 1 at $t = -2$ Comp = LossIn at $t = -2$	8	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 1 at $t = -2$ fSN = 2 at $t = -2$ Comp = LossIn at $t = -2$
3	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 0 at $t = -2$ fSN = 1 at $t = -2$ Comp = Down at $t = -2$	9	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 1 at $t = -2$ fSN = 2 at $t = -2$ Comp = Down at $t = -2$
4	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 0 at $t = -2$ fSN = 1 at $t = -2$ BFV = Frz at $t = -2$	10	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ SbnP = 1 at $t = -2$ fSN = 2 at $t = -2$ BFV = Frz at $t = -2$
5	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ fSN = 1 at $t = -2$ BFV = Arb at $t = -2$	11	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ fSN = 2 at $t = -2$ BFV = Arb at $t = -2$
6	L = 0 at $t = -2$ ELP = 0 at $t = -2$ CL = 0 at $t = -2$ fSN = 1 at $t = -2$ BFV = Zero at $t = -2$		

implicants shown in Table IX are deductively identified. Prime implicants #1 through #4 and prime implicants #7 through #11 identify that the conditions BFV failed stuck, computer loss of inputs, downing of the BC, or freezing of the BFV controller, together with a steam flow-feed flow mismatch (steam flow > feed flow) will lead to low level in the SG. This is because any of these failures will cause the feed flow to remain the same, while the steam flow slowly decreases. On the other hand, prime implicants #5 and #11 identify a condition corresponding to the BFV controller failing in the arbitrary state, whereas prime implicant #6 identifies a condition corresponding to the BFV controller failing in the zero state.

V.B.2. Inductive Analysis of the Benchmark System

Besides the deductive analysis, inductive failure and fault analyses were executed for the example initiating event. These inductive analyses identified the progression of system states from different combinations of initial component states potentially related to the system initiating event (please recall that at the whole system level, the term "initiating event" is here used in the PRA risk scenario/ET sequence sense).

As an inductive failure and fault analysis example, to identify the event sequence resulting from a stuck BFV, the following set of component initial conditions was used:

At time 0, BFV = F-S and remains in the same state AND

At time 0, CL = 0 AND

At time 0, CP = 0 AND

At time 0, Comp = OP and remains in the same state AND

At time 0, ELP = 0 AND

At time 0, LP = 0 AND

At time 0, SbnP = 0 AND

At time 0, fSN = 1 AND

At time 1, fSN = 1 AND

At time 2, fSN = 1 AND

At time 3, fSN = 1 .

Here F-S is the DFM state for the BFV failed stuck, CP is the DFM node for the compensated power, LP is the DFM node for the previous SG level, and SbnP is the DFM node for the previous BFV position.

These conditions correspond to the failure of the BFV in the stuck position while there is a mismatch between the steam flow and the feed flow (steam flow > feed flow). The DFM inductive analysis engine was then used to trace through the causality of the model, proceeding from the set of nodes whose states were set as initial conditions onward to downstream nodes, to determine the possible states of the latter. When the forward tracing is completed for one time step, the inductive engine updates node states according to the logic rules established by the time transition boxes/decision tables and any associated dynamic consistency constraints, all along applying the necessary logic reductions and manipulations. The intermediate steps of tracing through transfer box Tf3 and transfer box Tf1 are shown for illustration in Tables X and XI, respectively. In Tables X and XI, the columns in normal face represent the inputs to the transfer box in question, and the column in boldface represents the output for the same box. The first row indicates the time stamp associated with the input and output nodes. A time stamp of 0 indicates the initial time step, and it increases by 1 after a complete traversal of the loop. For example, in Table XI, given the input states (from the initial condition) ELP = 0 and CL = 0, the decision table for Tf3 was consulted to determine that the output state is EL = 0. This newly derived state of EL, together with the states of the nodes Comp, BFV, SbnP, and CP (defined in the initial condition) were used to determine the state of Sbn from the decision table associated with transfer box Tf1. This step is summarized in Table XI. After the inductive analysis has traced through all the transfer boxes, the forward tracing for time step 0 is completed. The next step is the forward tracing through the transition boxes. For example, Table XII shows the results of forward tracing through transition box Tt9. In summary, this inductive analysis showed that the BFV failure in the stuck position, together with an initial steam flow feed flow mismatch (steam flow > feed flow), will cause the SG level to drop from the normal state (L = 0), to the lowest state (L = −2) in two time steps, from LP = 0 at time step 0 (equivalent to L = 0 at time step −1) to L = −2 at time step 1. The final state of the SG level is shown in Table XIII as case 1.

TABLE X

Forward Tracing Through Transfer Box Tf3

Time	0	0	**0**
Node	ELP	CL	**EL**
State	0	0	**0**

TABLE XI

Forward Tracing Through Transfer Box Tf1

Time	0	0	0	0	0	**0**
Node	Comp	BFV	SbnP	CP	EL	**Sbn**
State	OP	F-S	0	0	0	**0**

TABLE XII

Forward Tracing Through Transition Box Tt9

Time	0	0	0	0	**1**
Node	L	fSN	Sbn	CL	**CL**
State	−1	1	0	0	**0**

TABLE XIII

Forward Tracing Through Transfer Box Tf2
(Inductive Analysis Case 1)

Time	1	1	1	**1**
Node	Sbn	fSN	LP	**L**
State	0	1	−1	**−2**

Of course, just as it may be deductively analyzed for a variety of separate and distinct top events, the system DFM model can also be inductively analyzed for many separate and distinct initial conditions of interest. For example, the DFWCS model can be analyzed for the initial condition set:

At time 0, BFV = Frz and remains in this state AND

At time 0, CL = 0 AND

At time 0, CP = 0 AND

At time 0, Comp = OP and remains in this state AND

At time 0, ELP = 0 AND

At time 0, LP = 0 AND

At time 0, SbnP = 2 AND

At time 0, fSN = 1 AND

At time 1, fSN = 1 AND

At time 2, fSN = 1 AND

At time 3, fSN = 1 .

Here, Frz is the DFM state of the BFV controller failed in the frozen state.

This set corresponds to the failure of the BFV controller in the frozen state while there is a mismatch between the steam flow and the feed flow (steam flow < feed flow). The automated inductive analysis of this scenario proceeds in the same basic fashion illustrated earlier. In summary, it shows that the BFV controller failure in the frozen state, together with an initial steam flow feed flow mismatch (feed flow > steam flow), will cause the SG level to rise from the normal state (L = 0), to the highest state (L = +2) in two time steps, from LP = 0 at time step 0 (equivalent to L = 0 at time step −1) to L = +2 at time step 1. The final state of the SG level is shown in Table XIV as case 2.

TABLE XIV

Forward Tracing Through Transfer Box Tf2
(Inductive Analysis Case 2)

Time	1	1	1	**1**
Node	Sbn	fSN	LP	**L**
State	2	1	+1	**+2**

V.B.3. Quantification of Benchmark System DFM Analysis Results

A dedicated multivalued logic quantification algorithm is used to quantify results obtained in a DFM deductive analysis. This algorithm is essentially the multivalued logic equivalent of binary decision diagram quantification schemes.[38] The DFM algorithm estimates the probability of the top event based on the probability estimates of the basic events that make up the TPI. If the deductive analysis has yielded n prime implicants, PI#1 through PI #n, as shown in Eq. (1),

$$Top\ Event = PI\ \#1 \vee \cdots \vee PI\ \#n \qquad (1)$$

$$PI\ \#i \subseteq PI\ \#j \ , \quad \text{for any } i \neq j \ ,$$

then this set of prime implicants is first converted into a set of m mutually exclusive implicants, MEI #1 through MEI #m, as shown in Eq. (2). These mutually exclusive implicants can be thought of as the multivalued logic equivalent of cut sets that do not yield any cross product term. Thus, the sum of the probabilities of these mutually exclusive implicants yields the probability of the top event, as shown in Eq. (3):

$$Top\ Event = \text{MEI}\ \#1 \vee \cdots \vee \text{MEI}\ \#m \ , \qquad (2)$$

where MEI #$i \wedge$ MEI #$j = \phi$ for any $i \neq j$ and

$$P(TopEvent) = P(\text{MEI}\ \#1) + \cdots + P(\text{MEI}\ \#m) \ . \qquad (3)$$

VI. SYSTEM FAILURE ANALYSIS USING THE MARKOV/CCMT METHODOLOGY

In the failure and reliability modeling of digital I&C systems using Markov/CCMT, the system failure probability (i.e., the probability that top events are reached) is evaluated throughout a series of discrete transitions within

the system and controlled variable state-space (CVSS). These discrete transitions take into account the following items:

item 1. the natural dynamic behavior of the system (i.e., mass and energy conservation laws)

item 2. the control laws

item 3. hardware/firmware/software states and their impact on the controlled/monitored process variables.

Items 1 and 2 are modeled using CCMT (Refs. 15 and 39). The hardware/firmware/software states referred to in item 3 are listed and defined in Table C.I.

Section VI.A describes the Markov model construction process. Section VI.B illustrates the generation of prime implicants from the Markov model that may need to be determined for the incorporation of Markov/CCMT results into an existing PRA.

VI.A. Benchmark System Markov/CCMT Model Construction

CCMT is a systematic procedure to describe the dynamics of both linear and nonlinear systems in discrete time and discretized system state-space (or the subspace of the controlled variables only). CCMT first requires a knowledge of the top events (Sec. VI.A.1) for the partitioning of the state space or the CVSS into V_j ($j = 1, \ldots, J$) cells (Sec. VI.A.2). The evolution of the system in discrete time is modeled and described through the probability $p_{n,j}(k)$ that the controlled variables are in a predefined region or cell V_j in the state-space at time $t = k\Delta t$ ($k = 0, 1, \ldots$) with the system hardware (such as pumps, valves, or controllers) and software/firmware having a state combination $n = 1, \ldots, N$ (Sec. VI.A.5). The state combination represents the system configuration at a given time and contains information regarding the operational (or the failure) status of each component (Sec. VI.A.3). Transitions between cells depend on (Sec. VI.A.4) (a) the dynamic behavior of the system, (b) the control logic of the control system, and (c) the hardware/firmware/software states.

The dynamic behavior of the system is usually described by a set of differential or algebraic equations, as well as the set of control laws, such as given in Appendix A. However, they can be any input-output relation, in general, including experimental data. The operating/failure states of each component are specified by the user. The procedure to determine the cumulative distribution function (Cdf) and the probability distribution function (pdf) of each top event follows several steps. These steps are explained in Secs. VI.A.1 through VI.A.6.

VI.A.1. Definition of the Top Events

The controller is regarded as failed if the water level in SGn ($n = 1,2$) rises above $+30$ in. and falls below -24 in. (Sec. III.A). Consequently, there are two top events:

1. $x_n < -24$ in. (low-level).
2. $x_n > +30$ in. (high level).

The cells that correspond to top events are modeled as absorbing cells or sink cells; i.e., the system cannot move out of these cells, and thus, the transition probabilities from these cells to others cells in the state-space or CVSS are equal to 0.

VI.A.2. Partitioning of the State-Space or the CVSS into Computational Cells

The dynamics of the system is modeled as transitions between cells V_j ($j = 1, \ldots, J$) that partition the state-space or CVSS. For the example initiating event, Eqs. (A.30) through (A.33) show that the CVSS is three-dimensional and comprises level x_n, level error E_{Ln} or BFV position S_{Bn}, and compensated level C_{ln}.

The partitioning needs to be performed in such a way that other than V_j being disjoint and covering the whole space (definition of partitioning), values of the controlled variables defining the top events (in our case x_n) and the setpoints (if any) must fall on the boundary of V_j and not within V_j. If this requirement is not satisfied for some V_j', then the system state becomes ambiguous when the state variables are within V_j' since the methodology assumes that $p_{n,j}(k)$ is uniformly distributed over V_j' (Refs. 15 and 39). Figure 20 shows a sample discretized CVSS based on Eqs. (A.30) through (A.33). Note that only three out of four variables in

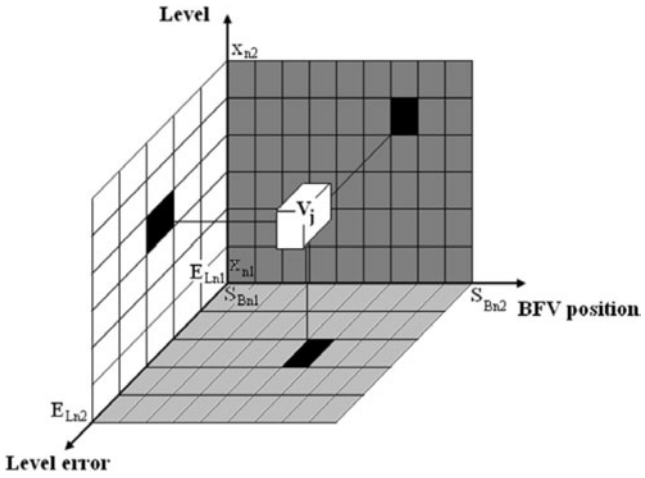

Fig. 20. The CVSS for the benchmark system based on Eqs. (A.30) through (A.33).

Eqs. (A.30) through (A.33) are independent, since $\tilde{S}_{Bn}(t)$ in Eq. (A.33) is a function of E_{Ln}.

VI.A.3. Definition of the Hardware/Firmware/Software States

The definition of states in the construction of the Markov models for the components listed in Sec. III.A follows the same conceptual reasoning presented in Appendix D for the construction of the finite state model of the benchmark system. The starting point is the FMEA presented in Table C.I. Each state identifies a specific status of the component under consideration, and transitions between different states belong to the failure states presented in Table C.I.

For the example event described in Sec. III.C, the relevant components are the BC and BFV controller. Then, from Figs. D.2 and D.3, the relevant states for the example initiating event are

1. BC operating and BFV controller operating
2. BC loss of inputs and BFV controller operating
3. BC down and BFV controller OK
4. freeze
5. arbitrary output
6. 0 dc voltage (vdc)
7. stuck.

VI.A.4. Determination of Hardware/Firmware/Software State Transition Probabilities

The stochastic behavior of hardware/software/firmware is represented through $h(n|n', j' \rightarrow j)$, which is the probability that the component state combination at time $t = (k+1)\Delta t$ is n, given that

Item 1. $n(k) = n'$ at $t = k\Delta t$.

Item 2. The controlled variables transit from cell V_j' to cell V_j during $k\Delta t \le t < (k+1)\Delta t$.

Item 2 reflects possible dependence of hardware/software/firmware state transitions on controlled variable transitions (e.g., setpoint crossings). For components with statistically independent failures, the probabilities $h(n|n', j' \rightarrow j)$ are simply the products of the individual component failure or nonfailure probabilities during the mapping time step from $k\Delta t$ to $(k+1)\Delta t$, i.e.,

$$h(n|n', j' \rightarrow j) = \prod_{m=1}^{M} c_m(n_m|n_m', j' \rightarrow j) \quad , \qquad (4)$$

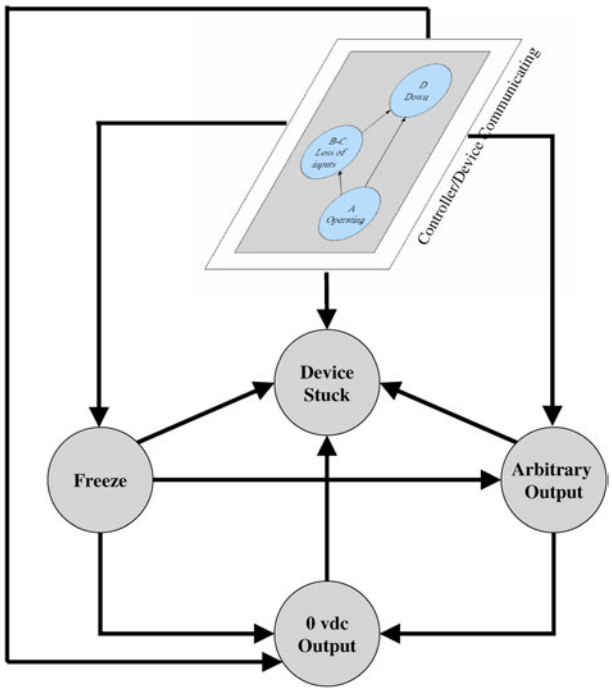

Fig. 21. Markov model of the hardware/software/firmware relevant to the example initiating event.

where $c_m(n_m|n_m', j' \rightarrow j)$ is the transition probability for component m from the combination n_m' to n_m within $[k\Delta t, (k+1)\Delta t]$ during the transition from the cell V_j' to V_j.

Figure 21 graphically illustrates the relevant benchmark DFWCS states and possible transitions between these states for the example initiating event based on Table C.I and Figs. D.2 and D.3. As an example of determining $h(n|n', j' \rightarrow j)$, suppose that the transition from the configuration n_m' to n_m in Eq. (4) involves the transition from the "Freeze" (see Appendix D) state (i.e., state 4) state to the "Arbitrary Output" state (i.e., state 5) with a failure rate equal to λ_{45}. Since there are only two components (i.e., BC and combined BFV-BFVcontroller), $m = 2$. Also, since the controller is in the Freeze state, BC is down, and the system meets the BFV demand at the most recent correct value (see Table C.I), which implies that the controller remains in the same state with probability $h(n|n', j' \rightarrow j) = \lambda_{45}\Delta t$.

VI.A.5. Determination of Cell-to-Cell Transition Probabilities

The cell-to-cell transition probabilities $g(j|j', n', k)$ are conditional probabilities that the controlled variables are in the cell V_j at time $t = (k+1)\Delta t$ given that (a) the controlled variables are in the cell V_j' at time $t = k\Delta t$ and (b) the system components are in component state combination $n(k) = n'$ at time t.

The $g(j|j',n',k)$ represents the dynamic behavior of the system as a function hardware/software/firmware states $n = 1,\ldots,N$ in discrete time and the discretized CVSS. They can be regarded as a probabilistic description of the dynamic evolution of the controlled variables under uncertainty of the system location in the CVSS (represented by V_j), possibly due to the discrete-time nature of the information sampled and model uncertainties. The $g(j|j',n',k)$ assumes that the system maintains its configuration j' within $k\Delta t \le t < (k+1)\Delta t$ and instantaneously moves to j at $t = (k+1)\Delta t$. The $g(j|j',n',k)$ can be determined from[39,40]

$$g(j|j',n',k) = \frac{1}{\nu_j}\int_{V_j} d\nu'\, e_j\{\tilde{\mathbf{x}}_{k+1}(\mathbf{x}',n',k)\} \qquad (5)$$

and

$$e_j\{\tilde{x}_k\} = \begin{cases} 1 \rightarrow \mathbf{x}_k \in V_j \\ 0 \rightarrow \text{otherwise} \end{cases} , \qquad (6)$$

where

$v_{j'}$ = volume of the cell $V_{j'}$

$\tilde{\mathbf{x}}_k$ = arrival point in the state-space/CVSS at time $t = (k+1)\Delta t$

$\mathbf{x}'$ = starting point in the cell $V_{j'}$ at time $t = k\Delta t$

n' = component state combination at time $t = k\Delta t$.

In Eq. (5), $\mathbf{x}$ is a vector whose components are the controlled variables (e.g., level x_n, level error E_{Ln}, compensated level C_{ln}, and BFV position S_{Bn} for the example initiating event described in Sec. III.C). The arrival point $\tilde{\mathbf{x}}_{k+1}(\mathbf{x}',n',k)$ is found from a given system model that describes system evolution as a function of system configuration [e.g., Eqs. (A.22) through (A.27)] with initial condition $\mathbf{x}(k\Delta t) = \mathbf{x}'$. As indicated above, the system configuration n' is assumed to be maintained during the determination of $\tilde{\mathbf{x}}_{k+1}(\mathbf{x}',n',k)$. The integral in Eq. (5) can be approximated by an equal-weight, N_p-point quadrature scheme using the following procedure:

1. Partition a cell j' into N_p equal size subcells.

2. Choose the midpoint of each subcell as initial conditions of Eqs. (A.14) through (A.19); integrate these equations over the time interval $k\Delta t \le t \le (k+1)\Delta t$ under the assumption that the component state combination remains n' at all times during $k\Delta t \le t \le (k+1)\Delta t$.

3. Observe the number of arrivals in N_{p+1} at time $t = (k+1)\Delta t$ [i.e., $\tilde{x}_k(x',n',k)$].

4. Obtain $g(j|j',n',k) = N_p/(N_{p+1})$.

VI.A.6. Construction of the Markov Model

The probability $p_{n,j}(k+1)$ $(j = 1,\ldots,J)$ that at $t = (k+1)\Delta t$ the controlled variables are in cell V_j and the component state combination is n can be found from[39,40]

$$p_{n,j}(k+1) = \sum_{n'=1}^{N}\sum_{j'=1}^{J} q(n,j|n',j',k)\,p_{n',j'}(k) \ , \qquad (7)$$

where

$$q(n,j|n',j',k) = g(j|n',j',k)\,h(n|n',j' \rightarrow j) \ . \qquad (8)$$

Since cells V_j cover the whole CVSS and N includes all the possible state combinations,

$$\sum_{n'=1}^{N}\sum_{j'=1}^{J} q(n,j|n',j',k) = 1$$

and

$$\sum_{n'=1}^{N}\sum_{j'=1}^{J} p_{n',j'}(k) = 1 \ . \qquad (9)$$

Note that for autonomous processes, the transition matrix $q(n,j|n',j',k)$ has to be constructed only once and not at each step throughout the duration of the mission of the system.

VI.B. Benchmark System Markov/CCMT Model Analysis

There are various possible ways the results from Eqs. (7) and (8) can be integrated into an existing PRA. For example, if the states and transitions in Fig. 21 are not relevant to the rest of the PRA and we are only interested in finding the top event probability, then

$$p_j(k) = \sum_{n=1}^{N} p_{n,j}(k) \qquad (10)$$

for j corresponding to low level or high level (Sec. VI.A.1) will give us this probability as a function of time and can be directly used in the PRA. However, if some states are common to other logical constructs (e.g., AND or OR gates) in the PRA, then the states need to be linked to these logical constructs. Reference 41 shows how the linkage can be performed by representing the prime implicants through dynamic ETs (DETs).

Reference 42 describes the construction of DETs from Markov/CCMT results. This section illustrates the process for the example initiating event in Sec. III.A. The basic idea of this approach is to use the transition matrix of the Markov model of the system as a graph representation of a finite state machine (a discrete process model of the stochastic dynamic behavior of the system). With

this representation and standard search algorithms,[43] it is possible to explore all possible paths to failure (scenarios) with associated probabilities and to construct DETs of arbitrary depth.

This section describes the DET analysis of the failure scenario detailed in Sec. III and the Appendixes and presents some results. Here is a summary of the assumptions made in Sec. III.C on the scenario under consideration:

1. Turbine trips.
2. Reactor is shut down.
3. Power $P(t)$ is generated from the decay heat.
4. Reactor power and steam flow rate (SFR) reduce to 6.6% of 3000 MW 10 s after the turbine trip.
5. Feedwater flow is at nominal level.
6. Off-site power is available.
7. MC is failed, and BC is in control.
8. FP fixed at minimum flow and does not fail.
9. MFV closed, and feedwater flow is controlled by the BFV.
10. There are two top events: low level and high level.

There are three independent process variables: level x_n, level error E_{Ln}, and compensated level C_{Ln} or BFV position $\tilde{S}_{Bn}(t)$ [see Eqs. (A.30) through (A.33)]. In addition, it is necessary to include the BFV position in the model to keep track of the position at which the BFV may be when/if it becomes stuck. The two system components controlling the process are the BC and the combined BFV-BFV controller. The BC can be in one of three distinct states (see Fig. 21):

1. Operating (OK)
2. Loss of inputs (LOSS/IN)
3. Down (DOWN).

The combined BFV and BFV controller can be in one of five distinct states (see Fig. 21):

1. Operating (OK)
2. Freeze: when it recognizes that BC is down (FREEZE)
3. Arbitrary output: a failure occurs inside the controller (ARB/OUT)
4. 0 vdc output: the signal from controller to valve is 0.0 (ZVDC/OUT)
5. Stuck: a mechanical failure of the valve occurs (STUCK).

For the purpose of this analysis, the following parameters were used (see Appendix D):

1. The water level x_n is partitioned into five intervals (all measures are expressed in feet) as shown in Table D.I.
2. The level error E_{Ln} is partitioned into three intervals (all measures are expressed in feet) as shown in Table D.II.
3. The compensated level C_{Ln} is partitioned into three intervals (all measures are expressed in feet) as shown in Table D.III.
4. The BVF position S_{Bn} is discretized into three intervals (percentage open) as shown in Table D.IV.

The time increment used was $\Delta t = 1$ s.

The number and size of the intervals to partition each process variable and the choice of the time increment Δt are bound by constraints described in Sec. VI.B. Essentially, a finer partition (with a larger number of smaller intervals) can yield a better approximation of the system at a cost of extra computational resources. Furthermore, the time increment is dependent on the size of the cells: Too small a time increment may result in the CCMT not producing useful results if most of the sample points and trajectories fail to leave the starting cell; too large an increment may cause some CCMT trajectories to cross multiple setpoints. Therefore, it is necessary to determine the partitioning scheme and the time interval by analyzing the actual system.

The partitioning chosen for the level variable is based on the following observations:

1. The LOW and HIGH points are identified in Sec. III.A.
2. Section III.A also points out that it is desirable to keep the level between ± 2 in. of the setpoint, i.e., ± 0.17 ft.

The other intervals for the level variable were added to provide a finer description of the behavior of the variable of primary interest.

The partitioning chosen for the BFV position is based on NUREG/CR-6465 (Ref. 13). The range of this variable is 0 to 100%. The range for level error and compensated level were determined experimentally through simulation of the system. The middle interval of the level error captures the entire range of values of the BFV position variable (which is computed as a function of level error). Finally, the partitioning for the compensated level was chosen to minimize the number of intervals while still modeling nominal, low, and high levels for this variable. Given the partitioning of the process variables, $\Delta t = 1$ s was chosen experimentally as an acceptable time increment relative to the size of the process variable intervals.

Figure 22 shows part of a DET generated for the example initiating event. The tool used to generate and display DETs starts from a normal state in which all the system components are operational and the process variables are within their nominal range. It then generates (employing a variant of algorithm 2 from Ref. 42) all possible configurations at the next time step (in this case 1 s), keeping track of all the possible states the process variables may be in at that point in time and in that configuration of the system components.

Figure 22 shows the tool window: The left pane shows a primitive representation of the ET, and the right pane shows the possible process states for the configuration and time step currently selected in the left pane. Instead of showing the events between branching points (as is usually done when displaying ETs), the representation of the ET in the left pane shows the configuration of the control units at each branching point. The event(s) corresponding to a specific branch in the tree can be deduced by comparing the configurations to the left and to the right of the branch. For instance, if in the configuration at the left of a branch both the BFV and the BC are operational (OK-OK) and in the configuration to the right the BC is down (OK-DOWN), the event that has occurred along that branch must be that the BC had a problem and took itself down. At each branching point (or node) in the ET, the node label shows the state of both the BFV and the BC.

The ET in the left pane is generated on demand. The top of the tree (displayed in the top-left corner of the left pane in Fig. 22) represents the normal configuration where all the system components are operational (OK-OK, i.e., the ensemble BFV and BFV controller and the BC are both operating correctly). Whenever the user clicks one of the displayed nodes (branching points), the program generates all the possible configurations in which the system may evolve in the given time step. For example, there are seven such possible distinct configurations after the first time step because from the OK-OK state the system can evolve into any of the seven states (see Fig. 21). By repeatedly clicking and expanding the tree nodes, the user can explore any possible scenario in the tree. For instance, the (partial) ET shown in Fig. 22 corresponds to one possible path (or failure scenario) leading to the level going below the LOW setpoint (dryout).

Boxes in the left pane of Fig. 22 highlight a possible failure scenario presented in detail in Table XV. The rounded box in the right pane shows the final failed state for this scenario: The value of each process variable (i.e., level, error level, compensated level, and BFV position) and the value of the SFR and feedwater flow rate (WFR). Note that in going from $t = 2$ s to $t = 3$ s, a minimum BFV aperture change of 40%/s (difference in the high BFV position of $S_{Bn} = 30\%$ at $t = 2$ s versus low BFV position of $S_{Bn} = 70\%$ at $t = 3$ s in Table XV) occurs. While this rate of change may be high, no inertia in the valve response

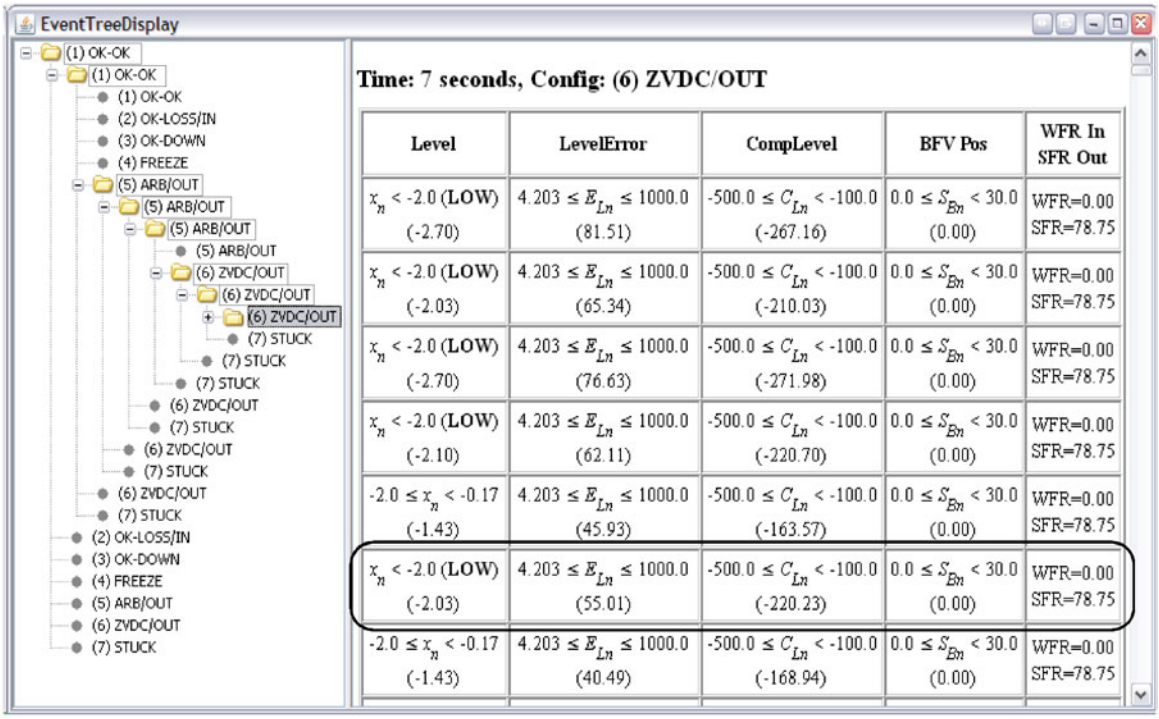

Fig. 22. Display of part of the DET.

TABLE XV

Example Failure Scenario

Time (s)	System Configuration	Process State	Explanation
$t = 0$	BFV: OK BC: OK	$-0.17 \le x_n < 0.17$ $-1.587 \le E_{Ln} < 4.203$ $-100.0 \le C_{Ln} < 100.0$ $0.0 \le S_{Bn} < 30.00$	Both BFV and BC are in their operational state, and all process variables are in their nominal range.
$t = 1$	BFV: OK BC: OK	$-0.17 \le x_n < 0.17$ $4.203 \le E_{Ln} \le 1000.0$ $-100.0 \le C_{Ln} < 100.0$ $70.0 \le S_{Bn} \le 100.0$	Level error is high, so BFV opens more.
$t = 2$	BFV: ARB/OUT BC: OK	$0.17 \le x_n < 2.5$ $4.203 \le E_{Ln} \le 1000.0$ $-100.0 \le C_{Ln} < 100.0$ $0.0 \le S_{Bn} < 30.0$	BFV controller fails and starts generating arbitrary outputs to the valve, in this case a low value. The level is higher than the nominal level interval.
$t = 3$	BFV: ARB/OUT BC: OK	$-2.0 \le x_n < -0.17$ $4.203 \le E_{Ln} \le 1000.0$ $-100.0 \le C_{Ln} < 100.0$ $70.0 \le S_{Bn} \le 100.0$	BFV controller is still generating arbitrary outputs, in this case a high value. The level is lower than the nominal level interval.
$t = 4$	BFV: ARB/OUT BC: OK	$0.17 \le x_n < 2.5$ $4.203 \le E_{Ln} \le 1000.0$ $-100.0 \le C_{Ln} < 100.0$ $0.0 \le S_{Bn} < 30.0$	BFV controller is still generating arbitrary outputs, in this case a low value. The level is higher than the nominal level interval.

was directly taken into account because of a lack of reliable data and because it does not affect the ultimate outcome of the scenario. The valve inertia is somewhat indirectly accounted for in the rate of level change, which from Table A.I and Fig. A.1 is 140/109.0 = 1.28 ft/s when the BFV is fully open.

The ET constructed for this analysis uses a modified version of algorithm 2 of Ref. 42. The CCMT employed to determine the possible behavior of the system starts with a set of sample points (27 in this specific case) located on a regular grid inside the process state-space cell representing all the variables being in their nominal range, i.e., $-0.17 \le x_n < 0.17$, $-1.587 \le E_{Ln} < 4.203$, and $-100.0 \le C_{Ln} < 100.0$. The algorithm then follows the evolution of the system through time and through changing configurations of system components (BFV and BC) by always starting the next cell-to-cell mapping from the locations within a cell where it landed at the previous time step. This allows the ET display tool shown in Fig. 22 to display not only the intervals in the state-space within which each variable is contained at any point in time for a given scenario but also information about the exact values of the variables. This information is displayed in the left pane below each variable interval (the number in parentheses in Fig. 22). Note that the tool displays also the current values of the feedwater inflow rate and of the steam outflow rate (the last column in the right pane in Fig. 22). This allows the user to know at once whether the level is going up (when WFR > SFR) or down (when WFR < SFR).

In addition to generating DETs with the tools described, it is also possible to use the same variant of algorithm 2 (Ref. 42) to compute complete DETs to a given depth. Table XVI summarizes the number of failure scenarios exhibited by the system as a function of the depth of the tree, i.e., the length of time for which the system is analyzed. The percentage of the total number of scenarios for a given depth that lead the system to fail LOW, fail HIGH, and not fail is included in parentheses.

As can be seen from Table XVI, there are a large number of possible scenarios. The majority of scenarios for each DET depth fail to lead the system to failure within the chosen time limit. However, there are still a substantial number of scenarios leading to failure. This is due in part to the presence in the model of a state (ARB/OUT) of the system where the BFV can receive an arbitrary signal from the controller. This is modeled by exploring scenarios for three different values of the BFV position, one for each of the three intervals in which the BFV position has been partitioned (Table D.IV). Another observation is that the number of LOW failure scenarios is always much larger than the number of HIGH

TABLE XVI

Number of Failure/Nonfailure Scenarios

Time (s) (Depth of DET)	Number of LOW Failure Scenarios	Number of HIGH Failure Scenarios	Number of Scenarios Without Failure
1	0 (0.0%)	0 (0.0%)	243 (100.0%)
2	0 (0.0%)	0 (0.0%)	1 242 (100.0%)
3	530 (10.8%)	0 (0.0%)	4 384 (89.2%)
4	1 480 (9.3%)	0 (0.0%)	14 439 (90.7%)
5	4 999 (10.2%)	186 (0.4%)	43 727 (89.4%)
6	14 811 (10.2%)	2 518 (1.7%)	127 292 (88.0%)
7	47 881 (11.5%)	6 531 (1.6%)	362 153 (86.9%)
8	140 644 (11.9%)	18 559 (1.6%)	1 022 695 (86.5%)
9	411 240 (12.3%)	50 259 (1.5%)	2 871 468 (86.2%)
10	1 126 498 (12.0%)	143 922 (1.5%)	8 091 530 (86.4%)

failure scenarios. This is due to the existence in the model of a state (ZVDC/OUT) in which the BFV is closed. Whenever the system enters this state, the valve is forced to close and never reopens. Finally, Table XVI shows that given the stated initial conditions, the minimum time necessary for the system to fail LOW is 3 s, and the minimum time for the system to fail HIGH is 5 s.

Given the large number of failure scenarios, it is unrealistic to examine them directly. Also, the user may or may not choose to use all these scenarios depending on how the system under consideration is connected to the other plant systems in the full system PRA. For example, if the level information for the example DFWCS is not being used by other plant systems, then only the hardware/software/firmware states are relevant. Then, it is possible to remove exact timing information and detailed information about the evolution of the process variables to reduce the large number of failure scenarios to a more manageable set of sequences of component failure events leading to a failure of the system using algorithm 2 of Ref. 42. For the model being considered, there are only 64 distinct sequences of component failures that are possible. Table XVII shows for each sequence and for different lengths of failure scenarios up to 10 s the number of failure scenarios that follow the given sequence of component failures.

For instance, let us consider LOW failure scenarios of 10-s length. Table IX states that there are 1 126 498 such scenarios. The column labeled "Low 10" of Table XVII shows how many of these scenarios follow each possible sequence of component failures. In particular, the most common sequence is "1-5" (148 269 different scenarios). This means that there are 148,269 LOW failure scenarios that result simply from the system going through configurations 1 and 5. Configuration 1 refers to the state in which both the BFV and the BC are operating normally, and it is the initial configuration in all scenarios analyzed. Configuration 5 is the one where the BFV controller has failed and it is essentially sending arbitrary signals to the valve.

As can be seen from Table XVII, there are 64 possible distinct sequences of system configurations that can occur. Of these, four sequences ("1," "1-2," "1-2-3," and "1-3," identified in boldface in Table XVII) do not result in the system failing (either HIGH or LOW) within the 10-s time interval considered in the analysis. The system does not fail when everything remains operational (scenario 1 in Table XVII) as expected. As long as the controller is functional, the system behaves properly. The system also does not fail within 10 s for scenario 1-2 or 1-2-3 because states 2 and 3 do not have self-loops (see Fig. 21) and hence are transitory. Whenever the system enters either of these states, it is guaranteed to abandon them at the next time step. Therefore, for sequences that terminate in state 2 or 3, the system must have been in state 1 (and operational) up until the last time step. It is conceivable that at some point in time, the system could fail (HIGH or LOW) within one time step after having transitioned from state 1 to states 2 or 3. But, this seems unlikely, and as noted, it does not happen within the time interval considered here.

Of the remaining 60 sequences of system configurations that can occur, Table XVII shows that 40 can result in the system failing HIGH or LOW depending on the exact timing of the events, and only 20 can result in the system failing LOW. The 20 sequences that cannot result in a HIGH failure (identified in italics in Table XVII) all have one thing in common: They represent scenarios in which state 6 (ZVDC/OUT) is reached when the level is still below the high setpoint. If that happens, the system cannot fail high because the BFV is closed entirely and the level immediately starts to go down. So, all the sequences containing state 6 cannot lead to a HIGH failure, except for sequences that end with state 6 by reaching that configuration once the level has already risen above the high setpoint. On the other hand, any sequence where

TABLE XVII

Classification of Failure Paths

Scenario (State Sequence)	Low 3	Low 4	Low 5	Low 6	Low 7	Low 8	Low 9	Low 10	High 5	High 6	High 7	High 8	High 9	High 10
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1-2-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1-2-3-4	8	9	9	9	13	26	34	39	2	21	23	24	36	54
1-2-3-4-5	0	3	3	3	51	321	1116	3573	9	138	243	462	1047	2577
1-2-3-4-5-6	0	0	0	0	32	223	903	3127	1	25	58	130	313	805
1-2-3-4-5-6-7	0	0	0	0	26	198	889	3296	0	0	0	0	0	0
1-2-3-4-5-7	0	0	0	0	31	179	667	2236	1	33	76	204	581	1658
1-2-3-4-6	0	1	1	1	15	50	101	165	2	21	23	24	36	54
1-2-3-4-6-7	0	0	0	0	29	86	192	388	0	0	0	0	0	0
1-2-3-4-7	0	1	1	1	4	21	41	57	3	46	52	56	98	164
1-2-3-5	24	30	36	87	306	1044	3441	10119	6	63	189	552	1434	3957
1-2-3-5-6	0	2	5	46	229	886	3009	9190	2	21	63	184	478	1319
1-2-3-5-6-7	0	0	1	33	206	908	3158	10092	0	0	0	0	0	0
1-2-3-5-7	0	2	4	42	174	652	2156	6440	3	46	124	405	1076	3111
1-2-3-6	8	9	10	23	35	69	101	130	0	0	0	0	0	0
1-2-3-6-7	0	1	2	36	65	164	260	387	0	0	0	0	0	0
1-2-3-7	8	9	9	9	13	26	34	39	2	21	23	24	36	54
1-2-4	8	9	9	9	13	26	34	39	2	21	23	24	36	54
1-2-4-5	24	33	39	90	357	1365	4557	13692	15	201	432	1014	2481	6534
1-2-4-5-6	0	2	5	46	261	1109	3912	12317	3	46	121	314	791	2124
1-2-4-5-6-7	0	0	1	33	232	1106	4047	13388	0	0	0	0	0	0
1-2-4-5-7	0	2	4	42	205	831	2823	8676	4	79	200	609	1657	4769
1-2-4-6	8	10	11	24	50	119	202	295	2	21	23	24	36	54
1-2-4-6-7	0	1	2	36	94	250	452	775	0	0	0	0	0	0
1-2-4-7	8	10	10	10	17	47	75	96	5	67	75	80	134	218
1-2-5	24	63	225	624	1962	5466	15750	42654	6	63	273	939	2670	7908
1-2-5-6	8	31	151	476	1602	4657	13614	37724	2	21	91	313	890	2636
1-2-5-6-7	0	10	96	381	1469	4557	13693	39041	0	0	0	0	0	0
1-2-5-7	8	23	118	355	1158	3263	9455	25920	5	67	249	817	2296	6859
1-2-6	8	10	27	38	65	100	137	151	0	0	0	0	0	0
1-2-6-7	8	11	54	95	180	292	422	470	0	0	0	0	0	0
1-2-7	8	9	9	9	13	26	34	39	2	21	23	24	36	54
1-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1-3-4	8	9	9	9	13	26	34	39	2	21	23	24	36	54
1-3-4-5	24	33	39	90	357	1365	4557	13692	15	201	432	1014	2481	6534
1-3-4-5-6	0	2	5	46	261	1109	3912	12317	3	46	121	314	791	2124
1-3-4-5-6-7	0	0	1	33	232	1106	4047	13388	0	0	0	0	0	0
1-3-4-5-7	0	2	4	42	205	831	2823	8676	4	79	200	609	1657	4769
1-3-4-6	8	10	11	24	50	119	202	295	2	21	23	24	36	54
1-3-4-6-7	0	1	2	36	94	250	452	775	0	0	0	0	0	0
1-3-4-7	8	10	10	10	17	47	75	96	5	67	75	80	134	218
1-3-5	24	63	225	624	1962	5466	15750	42654	6	63	273	939	2670	7908
1-3-5-6	8	31	151	476	1602	4657	13614	37724	2	21	91	313	890	2636
1-3-5-6-7	0	10	96	381	1469	4557	13693	39041	0	0	0	0	0	0
1-3-5-7	8	23	118	355	1158	3263	9455	25920	5	67	249	817	2296	6859
1-3-6	8	10	27	38	65	100	137	151	0	0	0	0	0	0
1-3-6-7	8	11	54	95	180	292	422	470	0	0	0	0	0	0
1-3-7	8	9	9	9	13	26	34	39	2	21	23	24	36	54
1-4	8	9	9	9	13	26	34	39	2	21	23	24	36	54
1-4-5	48	96	264	714	2319	6831	20307	56346	21	264	705	1953	5151	14442
1-4-5-6	8	33	156	522	1863	5766	17526	50041	5	67	212	627	1681	4760
1-4-5-6-7	0	10	97	414	1701	5663	17740	52429	0	0	0	0	0	0
1-4-5-7	8	25	122	397	1363	4094	12278	34596	9	146	449	1426	3953	11628
1-4-6	16	20	38	62	115	219	339	446	2	21	23	24	36	54
1-4-6-7	8	12	56	131	274	542	874	1245	0	0	0	0	0	0
1-4-7	16	19	19	19	30	73	109	135	7	88	98	104	170	272
1-5	42	222	783	2310	7158	20103	57354	148269	6	90	456	1716	5241	16047
1-5-6	24	162	637	1949	6055	17100	48680	128251	2	30	152	572	1747	5349
1-5-6-7	10	114	551	1821	5798	16558	47290	126897	0	0	0	0	0	0
1-5-7	24	129	467	1380	4237	11829	33505	87206	7	121	496	1708	5019	15089
1-6	14	32	54	67	93	134	163	180	0	0	0	0	0	0
1-6-7	24	73	134	181	264	399	491	547	0	0	0	0	0	0
1-7	8	9	9	9	13	26	34	39	2	21	23	24	36	54
Total	**530**	**1480**	**4999**	**14811**	**47881**	**140644**	**411240**	**1126498**	**186**	**2518**	**6531**	**18559**	**50259**	**143922**

state 6 (ZVDC/OUT) is reached while the level is below the high setpoint is bound to result in the system failing LOW exactly because the BFV is closed entirely.

The 40 sequences of system configurations that do not include state 6 (except possibly as the last configuration, reached once the system has already failed) can result in the system failing HIGH or LOW depending on the exact timing and BFV position at the time of failure. If the controller fails in any way, either the valve ends up being stuck at its old value (states OK-LOSS/IN, OK-DOWN, FREEZE, and STUCK), or it can take on arbitrary values (state ARB/OUT). In either case, if the resulting water inflow is greater than the steam outflow, the system will fail HIGH, and if the water inflow is lower than the steam outflow, the system will fail LOW (except for the case where the steam outflow manages to go below the water inflow *before* the system fails LOW—in that case, again, the system will end up failing HIGH). In principle, it could be possible for the system to hit state 5 (ARB/OUT) and go on without failure for an arbitrary amount of time. But, this is unlikely and would have to rely on the arbitrary output produced by the controller actually working to control the level successfully, a highly unlikely event. In any case, the analysis summarized in Table XVII clearly shows that both HIGH and LOW failures can occur whenever the system fails in state 5 (ARB/OUT).

The sample analysis presented in this section shows that it is possible to construct DETs from a Markov model of the system. The analysis can also produce qualitative information such as failure paths or ordered sequences of failure events.

VII. COMPARISON OF MARKOV/CCMT METHODOLOGY AND DFM ANALYSIS OF THE BENCHMARK SYSTEM RESULTS

It is useful to examine in comparative perspective the DFM and Markov/CCMT applications discussed in this paper as well as summarize the indications obtained in terms of a possible complementary use of the two methodologies, which was partially addressed at the beginning of Sec. IV.

The DFM was used for two separate analyses of the benchmark system: a deductive (or backward) analysis and an inductive (or forward) analysis. The deductive analysis resulted in two sets of prime implicants that could cause one of the two top events (level fails HIGH or LOW). Examination of the prime implicants showed that for the system and scenario under consideration, any failure of the BC or of the combined BFV-BFV controller could result in failure of the system. In particular, any system failure except 0 vdc output of the combined BFV-BFV controller could result in a high-level failure if the failure occurs when the feed flow is greater than the steam flow, and any system failure could result in a low-level failure if the failure occurs when the feed flow is smaller than the steam flow. The inductive analysis showed how the DFM model can be used to investigate the behavior of the system once a certain combination of initial component states has been defined. The two sample sequences generated simply confirmed the results predicted by the deductive analysis, i.e., that if the system starts in a state where BFV is stuck, the system could fail high if the position of the BFV is such that the feed flow is lower than the steam flow, and it could fail LOW if the position of the BFV is such that the feed flow is greater than the steam flow.

The Markov/CCMT methodology was used to perform a forward or inductive analysis of the example initiating event. A Markov/CCMT model of the system was employed to generate all possible failure scenarios within ten time steps (10 s) from the initiating event. This analysis revealed the large number of ways in which the system can evolve leading to failure (level HIGH or LOW). It showed all the sequences of component failure events that can lead to each kind of failure. It also showed that any failure of a system component (BC or combined BFV-BFV controller) can result in failure of the whole system and that the exact timing of the component failure, in addition to failure mode, is what determines whether the overall system will fail HIGH or LOW.

Although the two methodologies currently present the results of their respective analyses in different forms so that a direct comparison cannot be performed, they clearly agree on the high-level, summary assessment of the system failure modes. From both analyses it follows that the example benchmark system can fail as a result of any system component failure. The DFM results emphasize the relative magnitude of feed flow versus steam flow at the time of the system component failure as the discriminant to decide which kind of failure will occur (HIGH or LOW); the Markov/CCMT analysis emphasizes that the exact timing of the system component failure events will determine the kind of system failure.

These two characterizations of the results coincide with each other in all cases except for two scenarios:

1. A system component can fail when the feed flow is lower than the steam flow, but before the system can fail LOW, the steam flow (which decreases with time) falls below the now constant feed flow resulting in the system actually failing HIGH.

2. The BFV controller fails in the arbitrary output state when the feed flow is lower than the steam flow, but because of the potentially erratic nature of the BFV controller signal, the feed flow becomes greater than the steam flow before the system fails LOW, and the system ends up failing HIGH.

Neither of these two scenarios is expressed explicitly by the prime implicants resulting from the DFM analysis. However, the Markov/CCMT model can generate failure scenarios that capture these behaviors of the system. The reason why these two scenarios were not identified explicitly by DFM is that (a) the DFM deductive analysis looks for the shortest path, in terms of time steps, that leads to the top event and (b) in this case it was limited to only two time steps. In fact, it should be pointed out that the two scenarios identified by the Markov/CCMT eventually evolve at an intermediate point in time into the conditions expressed in the DFM prime implicants.

It is worthwhile pointing out that there are also some differences in the modeling of the initiating event employed by the analyses in the two approaches.

First, the DFM model included the steam flow as a modeled, independent variable. This allowed for time compression; i.e., a time increment in the DFM analysis can represent an arbitrary large time interval that is determined by the time needed for the system to transition from one level interval to the next. The Markov/CCMT model, instead, used actual time and considered steam flow a dependent variable determined by Eq. (A.21) as a function of time. Time compression allows DFM to analyze the system for a potentially longer time interval, while the number of possible scenarios limits the depth of the DET generated by the Markov/CCMT approach and therefore the length of the time interval that can be explored with this model. However, time compression also eliminates the details of the many scenarios that are possible and thus may remove potentially useful information.

Second, the DFM model assumed that all failure states of the BC and combined BFV-BFV controller are sink states for these components, while Markov/CCMT used the state transition diagram in Fig. 21. This caused some discrepancies in the results. For example, DFM-generated prime implicants state that the BC experiencing loss of input or going down can result in failure. Markov/CCMT, however, only generated failure paths that must include at least one more configuration change (failure) after the BC experiences loss of input or goes down. That is because in the model described by Fig. 21, the BC states for loss of input and down are transitory states with no self-loop and the model forces the system to transition to some other state at the very next time step. This explains why failure paths such as [BC and BFV both OK] → [BC down] are not included in the Markov/CCMT analysis results but are captured by the DFM analysis prime implicants.

In summary, the DFM backward analysis produces a more concise description of the high-level failure behavior of the system. For certain systems, such description may be entirely satisfactory and more manageable than the much more detailed results produced by the Markov/CCMT approach forward analysis. For other systems or for particular initiating events, it may be necessary to obtain detailed information about all possible failure paths and exact timing of the events. In such cases, one may need to appeal to the full power of the Markov/CCMT approach. In fully general terms, as we have mentioned at the beginning Sec. IV, the best strategy appears to use initially the DFM deductive analysis power to partition the "search space" for a given top event of interest into an orderly set of scenario subcases that can be individually explored further via the more detailed analytical simulations of the Markov/CCMT analysis.

VIII. CONCLUSIONS

This paper describes a control system that incorporates the distinguishing features of digital I&C systems and that can be used as a benchmark to assess the capabilities of the methodologies proposed for the PRA modeling of such systems. The paper also shows that the Markov/CCMT and DFM can be used for the PRA modeling of the benchmark DFWCS in a manner that is compatible with an existing plant PRA model. While a quantification of top event probabilities was not performed in conclusive fashion as part of this study, the study explored quantification via generation of data via the use of fault injection experiments as described in Sec. 2.4 of NUREG/CR-6942 (Ref. 30). Within this context, the potential issue of common-cause-failure quantification can be addressed by obtaining "beta factors" via simultaneous multiple fault injections. Note that epistemic uncertainties (such as modeling uncertainties or uncertainties in the initial conditions) can be accounted for in the quantification process via (a) the use of discrete state representation of the process dynamics and (b) use of multiple process trajectories sampled over the epistemic uncertainties to determine the transition probabilities between these discrete states [e.g., via Eqs. (5) and (6)].

Although the DFWCS is an effective benchmark system, the authors recognize that it does not have all the features associated with current and future digital systems planned for use in nuclear power plant applications. Especially in the perspective of addressing potential safety and regulatory priorities, it would be desirable also to define and investigate an additional complementary benchmark using a digital reactor protection system.

Finally, it should be mentioned that both DFM and Markov/CCMT have been applied to produce probabilistic risk estimates for initiating events of interest for the DFWCS benchmark system, including the one defined in Sec. III.C. Because these results rely on the probabilistic data for hardware and software component failures that are still quite preliminary, they are not presented here but will be presented in a future publication.

APPENDIX A

THE CONTROL LAWS FOR THE BENCHMARK DFWCS

The control laws for the feedwater controller for SGn (n = 1,2; see Fig. 1) under normal system operation can be expressed as follows:

Level:
$$\frac{dx_n}{dt} = A(f_{wn} - f_{sn}) \tag{A.1}$$

Flow demand:
$$C_{Fn}(t) = \beta_{Fn}(f_{sn}) \int dt[r_n - C_{Ln}(t) + E_{Fn}(t)] - \lambda_{Fn}(\sigma_{Bn}) \tag{A.2}$$

Compensated water level:
$$\tau_2 \frac{dC_{Ln}}{dt} = -C_{Ln}(t) + x_n + \tau_1(f_{wn} - f_{sn}) \tag{A.3}$$

Compensated flow error:
$$\tau_6 \frac{dE_{Fn}}{dt} + E_{Fn}(t) = \tau_7 \left[\frac{df_{wn}}{dt} - \frac{df_{sn}}{dt} \right] \tag{A.4}$$

BFV demand:
$$C_{Bn}(t) = \upsilon_{Bn}\alpha_M + \upsilon_{Bn} C_{pn}(t) + \beta_{Bn}(h_{wn}) \int dt[r_n - C_{Ln}(t)] - \lambda_{Mn}(\sigma_{Mn}) \tag{A.5}$$

Compensated power:
$$\tau_4 \frac{dC_{pn}}{dt} = -C_{pn}(t) + p_n + \tau_3 \frac{dp_n}{dt} \tag{A.6}$$

FP demand:
$$\sigma_{Fn}(t) = \begin{cases} \sigma_{Fn} & \text{If high-power operation} \\ \sigma_{Fn}(\max(C_{Fn}, \sigma_{Mn}^{-1}(C_{Fn})) & \text{If low-power operation} \end{cases} \tag{A.7}$$

MFV demand:
$$\sigma_{Mn}(t) = \begin{cases} \sigma_{Mn}(C_{Fn}) & \text{If high-power operation} \\ 0 & \text{If low-power operation} \end{cases} \tag{A.8}$$

BFV demand:
$$\sigma_{Bn}(t) = \begin{cases} 0 & \text{If high-power operation} \\ C_{Bn}(t) & \text{If low-power operation} \end{cases} \tag{A.9}$$

FP speed:
$$\tilde{S}_{Fn} = \begin{cases} \sigma_{Fnm} & \text{MC operational} \\ \sigma_{Fnb} & \text{MC failed, BC operational} \\ \eta_{Fn} & \text{MC failed, BC failed} \end{cases} \tag{A.10}$$

MFV position:
$$\tilde{S}_{Mn} = \begin{cases} \sigma_{Mnm} & \text{MC operational} \\ \sigma_{Mnb} & \text{MC failed, BC operational} \\ \eta_{Mn} & \text{MC failed, BC failed} \end{cases} \tag{A.11}$$

BFV position:
$$\tilde{S}_{nB} = \begin{cases} \sigma_{Bnm} & \text{MC operational} \\ \sigma_{Bnb} & \text{MC failed, BC operational} \\ \eta_{Bn} & \text{MC failed, BC failed} \end{cases} \tag{A.12}$$

PDI decision:
$$\tilde{S}_{Pn} = \begin{cases} 0 & \hat{S}_{Mn} > 0 \\ \eta_{Bn} & \text{otherwise} \end{cases}. \tag{A.13}$$

All sensor inputs are averaged before being used by the control laws. For example, the feedwater level for SG1 is the average of the two feedwater level sensors LV1 and LV2 (see Figs. 1 and 6).

In Eq. (A.1) the water inflow rate f_{wn} into SGn (see Fig. 2) depends on the MFV and BFV positions and FP speed, respectively, in general. The steam flow rate f_{sn} is determined from the physical process equations modeling the mass and energy transfer in SGn as modeled by the procedure described in Ref. 13. Equations (A.2), (A.3), and (A.4) compute the flow demand for the high-power mode for the feedwater controller. The $\beta_{Fn}(f_{sn})$, $\beta_{Bn}(h_{wn})$, $\lambda_{Fn}(\sigma_{Bn,})$, $\lambda_{Mn}(\sigma_{Mn})$ in Eqs. (A.2) and (A.5) are obtained from table lookups. The subscripts m and b in Eqs. (10), (11), and (12) refer to signals from the MC and BC, respectively. The η_{Fn}, η_{mn}, and η_{Bn} in Eqs. (A.10) through (A.13) denote history data for the FP, MFV, and BFV positions, respectively. If both of the MC and BC are failed, these data are used to determine the FP, MFV, and BFV positions.

For the example initiating event described in Sec. III.C, the system is in the low-power mode with power being generated by the decay heat, and subsequently only the BFV is being utilized (see Sec. III.A). Then, Eqs. (A.1) through (A.13) reduce to Eqs. (A.14) through (A.19) as the control laws for SGn ($n = 1,2$) under normal operating conditions:

Level:
$$\frac{dx_n}{dt} = A(f_{wn} - f_{sn}) \ , \tag{A.14}$$

Level error:
$$\tau_5 \frac{dE_{Ln}}{dt} = r_n - C_{Ln}(t) \ , \tag{A.15}$$

Compensated water level:
$$\tau_2 \frac{dC_{Ln}}{dt} = -C_{Ln}(t) + x_n(t) + \tau_1(f_{wn} - f_{sn}) \ , \tag{A.16}$$

Compensated power:
$$C_{pn}(t) = p(0)e^{-t/\tau_4} + \frac{(1+\tau_3)}{\tau_4}\int_0^t du\, p(t-u)e^{-u/\tau_4} \ , \tag{A.17}$$

BFV demand:
$$\sigma_{Bn}(t) = \mu_{Bn}\alpha_{Bn} + \mu_{Bn}C_{pn}(t) + \beta_{Bn}(h_{wn})E_{Ln}(t) \ , \tag{A.18}$$

and

BFV position (%):
$$\tilde{S}_{Bn} = \begin{cases} \sigma_{Bn} & \text{main or backup central processing unit (CPU) up} \\ \eta_{Bn} & \text{both main and backup CPU down} \ . \end{cases} \tag{A.19}$$

In Eq. (A.14), $f_{wn} = 0$ if BFV is failed closed. Otherwise, f_{wn} is obtained from the solution of

$$\frac{4.73L(100/\tilde{S}_{Bi})^2 f_{wn}^{1.852}}{C^{1.852}D^{4.87}} = 136 + 6.3 \times 10^{-6} f_{wn} - 4.6 \times 10^{-11} f_{wn}^2 \ , \tag{A.20}$$

where

D = diameter of inlet pipe to the BFV (ft)

L = fitting parameter

and f_{wn} is in cubic feet per second. Equation (A.20) uses the pump and valve models given in NUREG/CR-6465 (Ref. 13) and assumes that the pump head is equal to the head loss in the valve. In general, the steam flow rate f_{sw} can be obtained by the procedure given in NUREG/CR-6465 (Ref. 13). For the example initiating event, it is assumed that steam generation rate f_{sn} follows the decay heat generation rate, i.e.,

$$f_{sn}(t) = 0.066 \times 2102.8 \times \left[\frac{1}{(10+t)^{0.2}} - \frac{1}{(3.15\times 10^7 + t)^{0.2}}\right] . \tag{A.21}$$

Equation (A.21) is taken from Ref. 44, and time t is in seconds. In addition to assumptions 1 through 5 in Sec. III.C, Eq. (A.21) assumes that $f_{sn}(0) = 2102.8$ ft^3/s, the reactor has operated for 1 yr, and the starting point of the analysis is 10 s after the turbine trip. For the data in Table A.I, Eqs. (A.14) through (A.19) become

$$\frac{dx_n}{dt} = \frac{f_{wn}(\tilde{S}_{Bn})}{109} - \frac{0.066 \times 2102.8}{109} \times \left[\frac{1}{(10+t)^{0.2}} - \frac{1}{(3.15\times 10^7 + t)^{0.2}}\right] , \tag{A.22}$$

$$\tau_5 \frac{dE_{Ln}}{dt} = -C_{Ln}(t) \ , \tag{A.23}$$

$$\tau_2 \frac{dC_{Ln}}{dt} = -C_{Ln}(t) + x_n(t) + \tau_1 f_{wn}(\tilde{S}_{Bn}) - 0.066 \times 2102.8\tau_1 \times \left[\frac{1}{(10+t)^{0.2}} - \frac{1}{(3.15 \times 10^7 + t)^{0.2}} \right] , \quad \text{(A.24)}$$

TABLE A.I

Data Used for the Example Initiating Event

Variable	Value
$fs(0)$	0.066×2102.8 ft^3/s
$x(0)$	0 ft
$E_{Ln}(0)$	0 ft
$C_{Ln}(0)$	0 ft
$C_{pn}(0)$	0.066×1500 MW(thermal)
$P(0)$	0.066×1500 MW(thermal)
S_{Bn}	0%
L/D	2
C	140
D	0.5 ft
μ_{Bn}	1/15
a_{Bn}	0
r_n	0
b_{Bn}	$100 \times 12/54$
A	1/109.0 MW(thermal)$^{-1}$

$$C_{pn}(t) = 0.066 \times 1500 e^{-t/\tau_4} + 0.066 \times 1500 \frac{(1+\tau_3)}{\tau_4} \int_0^t du \times \left[\frac{1}{(10+u)^{0.2}} - \frac{1}{(3.15 \times 10^7 + u)^{0.2}} \right] e^{-(t-u)/\tau_4} , \quad \text{(A.25)}$$

$$\tilde{S}_{Bn}(t) = C_{pn}(t)/15 + 1200E_{Ln}(t)/54 , \quad \text{(A.26)}$$

and

$$\frac{146.53 f_{wn}^{1.852}}{\tilde{S}_{Bn}^2} = 136 + 6.3 \times 10^{-6} f_{wn} - 4.6 \times 10^{-11} f_{wn}^2 . \quad \text{(A.27)}$$

Figure A.1 shows that the solution of f_{wn} as a function of $\tilde{S}_{Bn}$ can be represented by the quadratic function

$$f_{wn} = 0.0014\tilde{S}_{Bn}^2 + 1.2681\tilde{S}_{Bn} - 1.2019 . \quad \text{(A.28)}$$

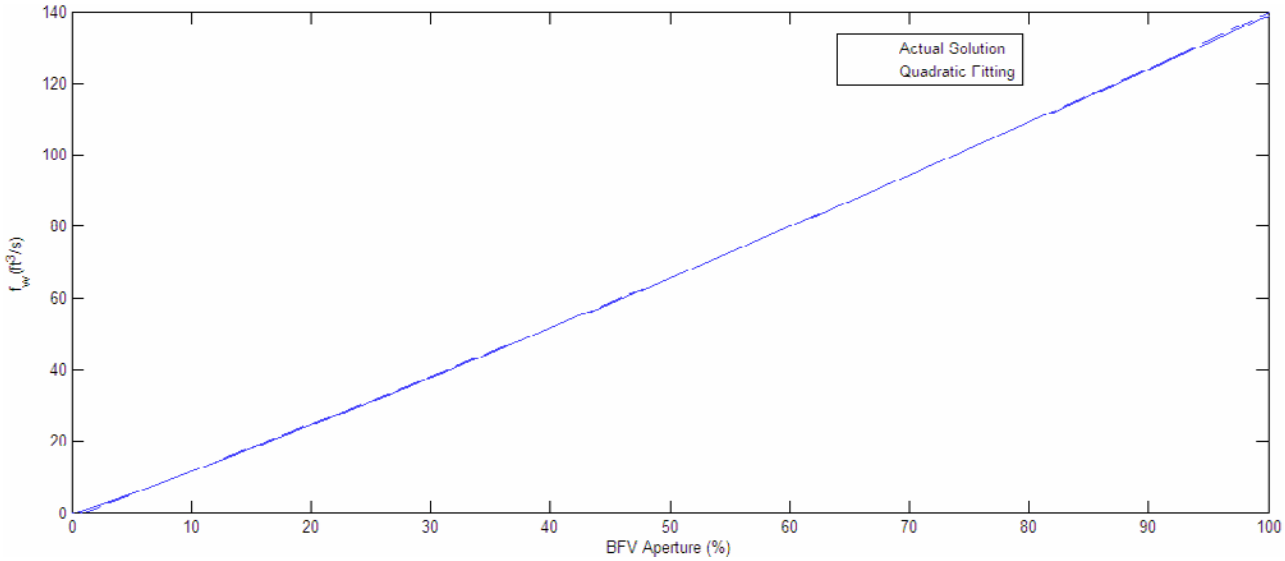

Fig. A.1. The solution of f_{wn} from Eq. (A.28) as a function of $\tilde{S}_{Bn}$.

Also, since

$$\frac{1}{(10+t)^{0.2}} \gg \frac{1}{(3.15\times 10^7+t)^{0.2}} \ , \qquad \text{(A.29)}$$

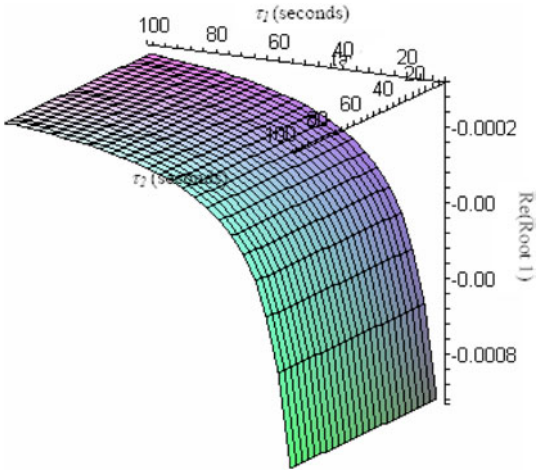

Fig. A.2. Real part of root 1 of the transfer function of Eqs. (A.30) through (A.33) following linearization around $E_{Ln} = 0$.

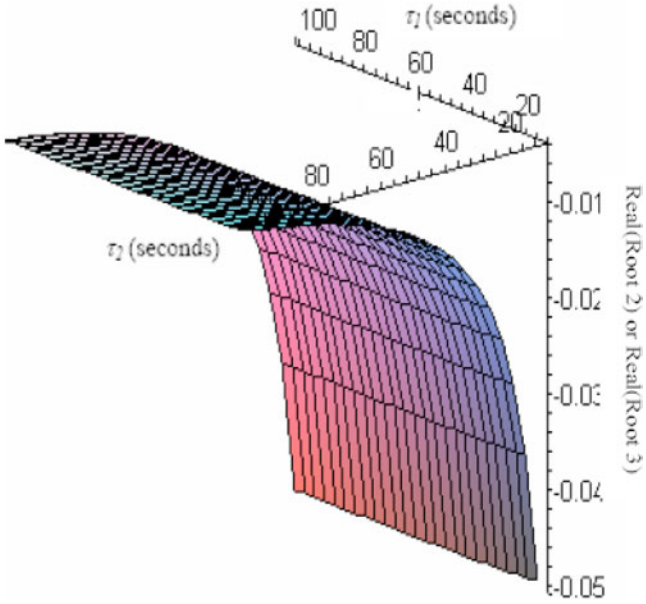

Fig. A.3. Real part of root 2 or root 3 of the transfer function of Eqs. (A.30) through (A.33) following linearization around $E_{Ln} = 0$.

we have

$$\frac{dx_n}{dt} = \frac{0.0014\tilde{S}_{Bn}^2 + 1.2681\tilde{S}_{Bn} - 1.2019}{109} - \frac{0.066\times 2102.8}{109}\frac{1}{(10+t)^{0.2}} \ , \qquad \text{(A.30)}$$

$$\tau_2 \frac{dC_{Ln}}{dt} = -C_{Ln} + x_n(t) + \tau_1(0.0014\tilde{S}_{Bn}^2 + 1.2681\tilde{S}_{Bn} - 1.2019) - \frac{0.066 * 2012.8\tau_1}{(10+t)^{0.2}} \ , \qquad \text{(A.31)}$$

$$-\tau_5 \frac{dE_{Ln}}{dt} = C_{Ln} \ , \qquad \text{(A.32)}$$

and

$$\tilde{S}_{Bn}(t) = 1/15\left[0.066 * 1500e^{-t/\tau_4} + 0.066 * 1500\frac{(1+\tau_3)}{\tau_4}\int_0^t du\,\frac{e^{-(t-u)/\tau_4}}{(10+u)^{0.2}}\right] + 1200E_{Ln}(t)/54 \ . \qquad \text{(A.33)}$$

The Laplace transform of Eqs. (A.30) through (A.33) following linearization around $E_{Ln} = 0$ shows that the transfer function has one real (root 1) and two complex (root 2 and root 3) conjugate roots. Figures A.2 and A.3 show that the system is unconditionally stable for $10 \le \tau_1 \le 100$ s and $10 \le \tau_2 \le 100$ s.

APPENDIX B

FAULT-TOLERANT FEATURES OF THE BENCHMARK DFWCS

The benchmark system has a number of fault-tolerant features:

1. The MFV, BFV, and FP controllers forward the control signals to the corresponding control points (the MFV, BFV, and FP, respectively). Thus, they provide a level of fault tolerance if both computers fail by allowing the operators time to intervene by holding the outputs of each to a previously valid value.

2. The computers, MFV and BFV and FP, and PDI controllers are each connected to an independent power source wired to a separate bus. A single power source failure can affect only one computer, all of the MFV/BFV/FP controllers, or the PDI controller at one time.

Both the MC and BC are set to oversample at three times the Nyquist criterion[f] to avoid aliasing.

3. The computers are able to process the sensor inputs and perform the control algorithms within one-third of the needed response frequency of the physical process. A failure in the MC or BC can be detected, and the failover[g] to a healthy component can occur with enough time to meet the response requirements of the process.

4. The water level setpoint is taken from a switch connected to the MFV and is propagated to all computers. If the setpoint signal goes out of range, then the computers fall back on a preprogrammed setpoint value.

5. Each computer is connected to a watchdog timer. In the case of a computer failure, the MFV, BFV, and FP controllers are notified and transfer control away from the affected computer.

6. Each computer verifies and validates its inputs, checking for out range and excessive rate changes in the inputs that would indicate errors in the sensor readings or problems with the analog-to-digital conversion of the values. Each computer will ignore input that fails these checks if the other inputs are still valid.

7. The values of the inputs are averaged across redundant sensors.

8. Deviation between the two sensors is detected, and if the deviation is large enough, the computer can signal a deviation error to the MFV, BFV, and FP controllers so they may switch to another computer.

9. The PDI controller provides one more level of fault tolerance, in that it holds the MFV to the needed position if the MFV controller does not produce output.

The DFWCS failover logic consists of the following. The MC has control of the control points initially, with the BC in hot standby. If the MC fails, then the BC takes control. If the BC fails after the MC has failed, then the MFV, BFV, and FP controllers each use one of their recent output values from the computer (essentially the last one that the controller can store) and recycle that value to the control points.

APPENDIX C

FAILURE MODES OF THE BENCHMARK DFWCS

As shown in Fig. 2, each feedwater controller consists of the MFV, BFV, PDI, FP, and their respective controllers, MC, BC, and sensors. Table C.I shows the failure modes of each component and their effects. "Out of range" failure mode for the sensor implies that the sensor drifts either high or low. In the "Stuck" mode, the MFV or BFV maintains its current position. While it is possible for either of the valves to fail without reaching its target position after it is actuated, this situation is not considered because of lack of data. The "Stuck" mode partially accounts for such a situation for the MFV or BFV. The "Stuck" mode for the FP implies that the FP maintains its current speed. As indicated in Appendix B (items 6 and 8), the computer's MFV, BFV, and FP controllers check their inputs for range and rate of change, providing the ability to detect failures in the MC and BC as well as the sensor data propagated to them. The failure mode "Operating but not able to detect failures" in Table C.I indicates the loss of this capability. The "Down" state for the MFV, BFV, and FP controllers implies that the MFV, BFV, and FP controllers have experienced an unrecoverable failure.

APPENDIX D

DISCRETE-STATE REPRESENTATION OF THE BENCHMARK DFWCS

A discrete-state representation of the benchmark DFWCS is more convenient for both the Markov/CCMT and DFM methodologies under consideration in this paper. For such a representation, the DFWCS topology can be regarded as consisting of three layers of interactions:

1. intracomputer interactions
2. intercomputer interactions
3. computer-controller-actuated device interactions.

The intracomputer interaction layer consists of five states (see Fig. D.1). In state A, the computer is operating correctly and nominally. In state B, the computer detects loss/invalid output for one sensor of any type (e.g., water level). State C represents loss/invalid output for two sensors of any one type. In state D the computer has detected an internal problem and is signaling that it has to be ignored. In state E, either the sensor output is invalid, or there is an internal processing error in the computer; however, the computer does not detect the fault and transmits the wrong information to the controllers.

The intercomputer interaction layer can be thought of as including the possible transfers of control of the actuated devices among the MC, BC, and controller. For example, the transfer of control from the MC to the BC would be represented here. There are three such computer-computer macro states (Fig. D.2). In state 1, both MC and BC are operating normally. In state 2, one computer is down but can be recovered. In state 3, again one computer is down, but it is not recoverable. Transitions between the macro states (MSs) depend upon the state of the controlling computer as shown in Fig. D.2.

[f] The Nyquist criterion states that the highest frequency present in a signal must be less than half of the sample frequency.[45]

[g] Failover is the process in which a degraded component is removed from control and replaced by a healthy component.

TABLE C.I

Benchmark System Component FMEA

Component	Failure Mode	Effect on the System
Sensor	Out of range	MC and BC use old values in their computations.
	Loss of output	No input to MC and BC.
MFV	Stuck	MFV maintains its position.
MFV controller	Loss of input	MFV controller performs failure over operations as necessary.
	Loss of output	If detected by PDI, old signal for the MFV is used. If undetected, the MFV position will decrease as no signal will be received.
	Operating but not able to detect failures	MFV controller will be unable to detect failures of either the MC or BC.
	Down	MFV controller may output any signal.
BFV	Stuck	BFV maintains its position.
BFV controller	Loss of input	BFV controller performs failure over operations as necessary.
	Loss of output	BFV position will decrease as no signal will be received from the FV controller.
	Operating but not able to detect failures	BFV controller will be unable to detect failures of either the MC or BC.
	Down	BFV controller may output any signal.
FP	Stuck	FP maintains its speed.
FP controller	Loss of input	FP controller performs failure over operations as necessary.
	Loss of output	FP speed will decrease as no signal will be received from the FP controller.
	Operating but not able to detect failures	FP controller will be unable to detect failures of either the MC or BC.
	Down	FP controller may output any signal.
PDI controller	Loss of input	PDI outputs the last signal it received from the MFV controller to the MFV.
	Loss of output	PDI is unable to mitigate a failure of the MFV.
	Down	FP may output any signal to the MFV.
MC	Loss of one input	MC uses old values to compute valve and pump demands.
	Loss of two inputs	MC uses old values to compute valve and pump demands.
	Intermittent failure	If the BC is OK, system transitions control to the BC. Otherwise, the system maintains valve and pump demand.
	Down	MC may output any signal and also will signal to the MFV, BFV, and FP controller that it has failed.
BC	Loss of one input	BC uses old values to compute valve and pump demands.
	Loss of two inputs	BC uses old values to compute valve and pump demands.
	Intermittent failure	BC may output any signal and also will signal to the MFV, BFV, and FP controller that it has failed.
	Down	If the MC is good, system transitions control to the MC. Otherwise, the system maintains the valve and pump demand.

Primary and secondary computers correspond, respectively, to the computer that is sending data to the controller and to the computer that is waiting in hot standby. Either the MC or BC can be the primary or the secondary computer. Recoverable and nonrecoverable failures are defined as follows:

1. Recoverable failure corresponds to the inability for the computer (which is still operating correctly) to send valid data to the controller (e.g., due to a loss of input from one or more sensors).

2. Nonrecoverable failure corresponds to an internal failure of the computer (e.g., the trip of the watchdog timer) or to a loss of output of the computer itself.

If the secondary computer (i.e., the computer that is not in control) fails and it is still recoverable, a transition from MS 1 to MS 2 occurs. These transitions simply take each state in MS 1 to the corresponding state in MS 2. For example, state A (or operational) in MS 1 would have a transition to state A in MS 2. When the secondary computer recovers, the opposite transitions occur (from MS 2

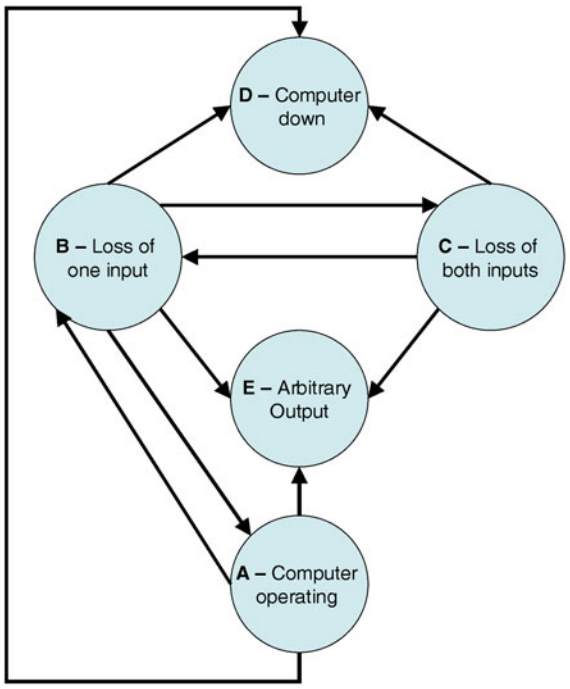

Fig. D.1. Intracomputer interactions.

to MS 1). Again, for example, if the operating computer is in state A, MS 1, then the transition would start there and end in state A in MS 1.

From state D in MS 1, the possible transitions represent the takeover of control of the process by the secondary computer (which from now on will be regarded as the primary computer). The transitions from this state go to all states except for state D in MS 2. The rationale behind these transitions is that the secondary computer was operating and may have transitioned to states other than state A in MS 2. The reason that state D in MS 2 is not a possible destination is that if state D is reached by the secondary computer, then another transition from MS 1 to MS 2 must have already taken that into account.

The failover action from MS 1 to MS 3 is a result of controller action via the watchdog timer or detecting the output failure from the computer. This action takes down the failed computer permanently and can occur in both the primary and secondary computers. If it occurs in the secondary, the transitions mimic the action of the secondary failure transitions from MS 1 to MS 3 by simply transitioning from a state in MS 1 to the respective state in MS 3. For example, state A in MS 1 would have a transition to state A in MS 3. If the primary computer

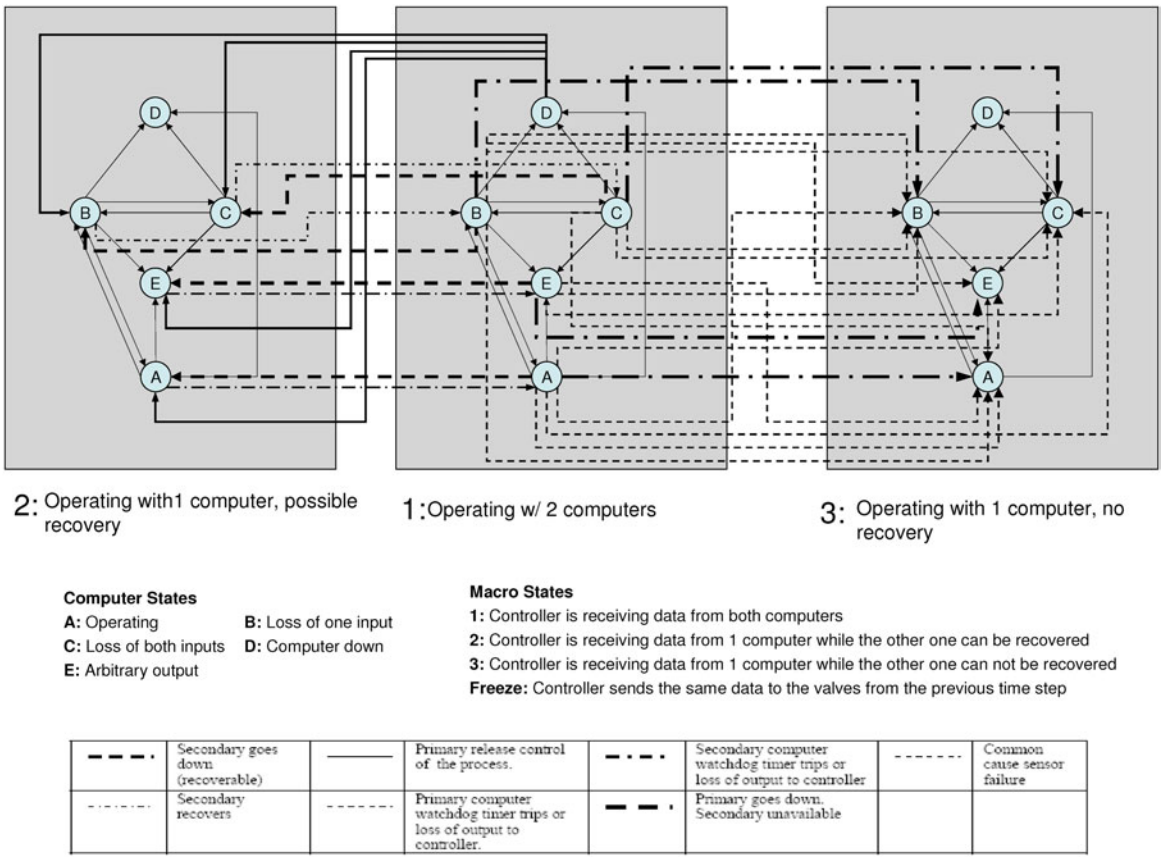

Fig. D.2. Intercomputer interactions.

fails in a nonrecoverable manner when both MC and BC are operating (i.e., when the DFWCS is in MS 1), then the DFWCS can go to any state in MS 3 except state D by the same rationale for transitions between MS 1 and MS 2. The transitions must take into account that the secondary computer may have already entered different states and these must be represented in the transitions to MS 3.

Figure D.3 shows all the possible controller-computer-actuated device interactions obtained from Table C.I. The shaded circles represent signals to the actuated devices (e.g., MFV, BFV, and FP) upon computer/controller failure, as well as the mechanical failure of the actuated device (device stuck). Mechanical failure of the actuated device leads to the device maintaining its current position for MFV and BFV or to zero flow for FP (see Table C.I). The planes represent the communication status between the controller and actuated devices. The two-way transitions between planes I and II are necessary to keep track of the computer from which the controller is receiving data when the communications between the controllers are restored.

As presented in Fig. D.3, the following types of controller failures are under consideration:

1. Arbitrary output: Random data are generated and sent to the actuated device (i.e., pump or valves).

2. Output high: Output value is stuck at the maximum value (i.e., valve totally open or pump at the maximum speed).

3. Output low: Output value is stuck at the minimum value (i.e., valve totally closed or pump stopped).

4. 0 vdc output: There is loss of communications between controller and actuated device.

Moreover, as a result of the failure of both computers, the controller can recognize the failure and send to the actuated devices (i.e., pump or valves) the old valid value (i.e., Freeze). If the controller does not recognize the failure, then it will simply pass on invalid information (arbitrary output) to the actuated device. Figure D.3 also shows how the computer-computer interactions (presented in Fig. D.2) integrate with computer-controller and controller-actuated device interactions. The behavior of the controller under normal and failed operation can be described as follows:

1. When both MC and BC are down, the controller transits to the freeze state. The actuated device remains in the position corresponding to the last valid value.

2. If the controller is operating and an output high or output low or arbitrary output failure occurs, the controller transits to the corresponding state, and the actuated device assumes the highest, the lowest, or an arbitrary position, respectively.

3. If the controller is in the freeze state and an output high, output low, or arbitrary output failure occurs, the controller transits to the corresponding state, and the

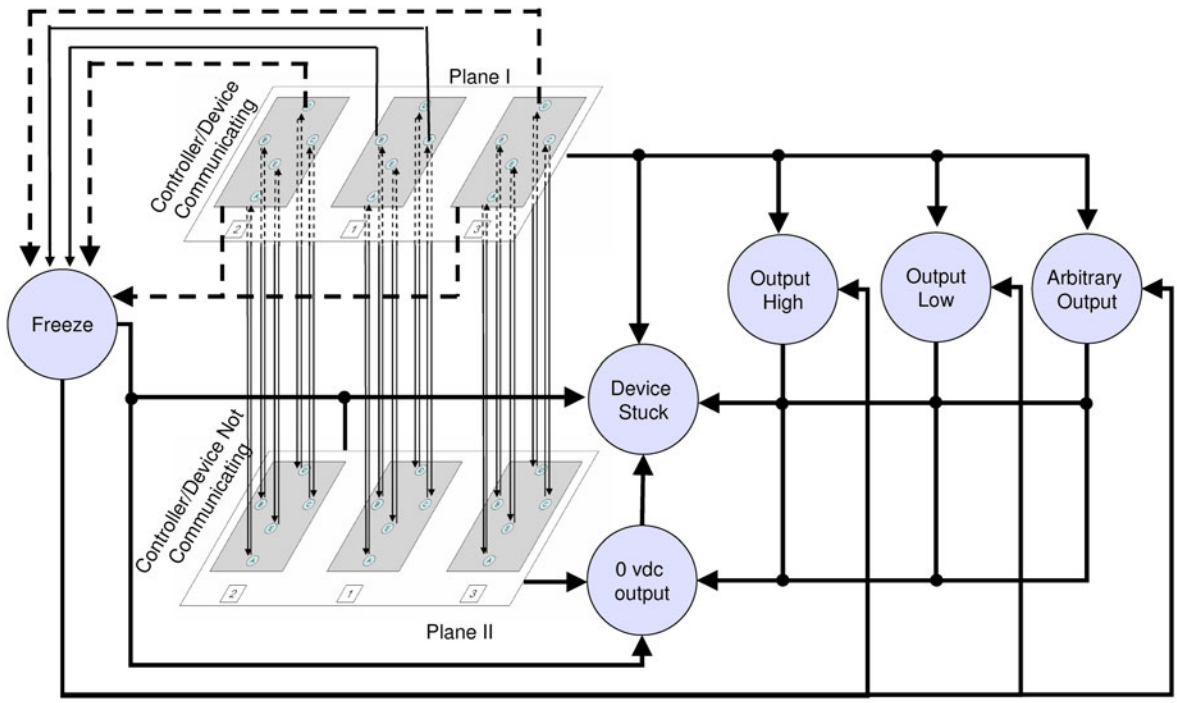

Fig. D.3. Computer-controller-actuated device interactions.

actuated device assumes the highest, the lowest, or an arbitrary position, respectively.

4. If a loss of output occurs when the controller is failed (i.e., the controller is sending arbitrary output, output high, or in the output low state), then the actuated device receives a 0 vdc as input, which correspond to the closed position or lowest speed.

Both the DFM and Markov/CCMT use a discrete state representation of the benchmark DFWCS. Tables D.I through D.IV show, respectively, the partitioning used for the water level, level error, compensated level, and BFV position in the modeling of the example initiating event described in Sec. III.C.

TABLE D.I

Partitioning of the Water Level x_n*

−2	$x_n < -2.0$ (LOW level)
−1	$-2.0 \le x_n < -0.17$
0	$-0.17 \le x_n < 0.17$
+1	$0.17 \le x_n \le 2.5$
+2	$x_n > 2.5$ (HIGH level)

*Expressed in feet.

TABLE D.II

Partitioning of the Level Error E_{Ln}*

−1	$-1000.0 \le E_{Ln} < -1.587$
0	$-1.587 \le E_{Ln} < 4.203$
+1	$4.203 \le E_{Ln} \le 1000.0$

*Expressed in feet.

TABLE D.III

Partitioning of the Compensated Level C_{Ln}*

−1	$-500.0 \le C_{Ln} < -100.0$
0	$-100.0 \le C_{Ln} < 100.0$
+1	$100.0 \le C_{Ln} \le 500.0$

*Expressed in feet.

TABLE D.IV

Partitioning of the BVF Position S_{Bn}*

−1	$0.0 \le S_{Bn} < 30.0$
0	$30.0 \le S_{Bn} < 70.0$
+1	$70.0 \le S_{Bn} \le 100.0$

*Percentage open.

NOMENCLATURE

C_{ln} = SGn compensated level

$c_m(n_m|n'_m, j' \to j)$ = Pr{component m is in state n_m at time $t = (k+1)\Delta t$| Component m is in state n'_m at time $t = k\Delta t$, and controlled/monitored variables move from cell j' to cell j during $k\Delta t \le t \le (k+1)\Delta t$}

E_{Ln} = SGn level error

$F_\gamma(k)$ = Cdf of top event γ at time k

f_{sn} = steam outflow rate for the SGn

f_{wn} = water inflow rate for the SGn

$g(j|j', n', k)$ = Pr{controlled/monitored variables are in cell j at time $t = (k+1)\Delta t$| controlled/monitored variables are in cell j' at time $t = k\Delta t$}

$h(n|n', j' \to j)$ = Pr{hardware/software/firmware in state n at time $t = (k+1)\Delta t$| hardware/software/firmware in state n' at time $t = k\Delta t$, and controlled/monitored variables move from cell j' to cell j during $k\Delta t \le t \le (k+1)\Delta t$}

J = total number of V_j

M = number of components

N = number of component state combinations

N_m = total number of n_m

n = component state combination index

n_m = component state index ($n_m = 1, \ldots, N_m$)

$P(t)$ = power

$p_{n,j}(k)$ = Pr{controlled variables are in cell j, and hardware/software/firmware is in state n at time $t = k\Delta t$}

$q(n, j|n', j', k)$ = elements of the transition matrix for the Markov chain

S_{Bn} = SGn BFV position

t = time

V_j = cells that partition the CVSS ($j = 1, \ldots, J$)

$w_{n,\gamma}(k)$ = pdf of top event γ at time k

x_n = water level of SGn

Greek

Γ = number of top events

γ = top event

Δt = cell-to-cell mapping time step

$\lambda_{n'_m, n_m}, \mu_{n'_m, n_m}$ = transition rates from component state n'_m to n_m

ACKNOWLEDGMENT

The research presented in this paper was sponsored by the U.S. Nuclear Regulatory Commission.

REFERENCES

1. "Use of Probabilistic Risk Assessment Methods in Nuclear Activities: Final Policy Statement," *Federal Register*, **60**, 43622 (1995).

2. T. ALDEMIR, D. W. MILLER, M. STOVSKY, J. KIRSCHENBAUM, P. BUCCI, A. W. FENTIMAN, and L. M. MANGAN, "Current State of Reliability Modeling Methodologies for Digital Systems and Their Acceptance Criteria for Nuclear Power Plant Assessments," NUREG/CR-6901, U.S. Nuclear Regulatory Commission (2006).

3. J. KIRSCHENBAUM, M. STOVSKY, P. BUCCI, T. ALDEMIR, and S. A. ARNDT, "Benchmark Development for Comparing Digital Instrumentation and Control System Reliability Modeling Approaches," *Proc. Int. Topl. Mtg. Probabilistic Safety Analysis (PSA '05)*, San Francisco, California, September 11–15, 2005, American Nuclear Society (2005) (CD-ROM).

4. H. KUMAMOTO and E. J. HENLEY, "Top-Down Algorithm for Obtaining Prime Implicant Set of Non-Coherent Fault Trees," *IEEE Trans. Reliab.*, **R-27**, 242 (1977).

5. T. ALDEMIR and N. O. SIU, "Guest Editorial," *Reliab. Eng. Syst. Safety*, **52**, 181 (1996).

6. D. JOHNSON, *The Sampling Theorem*; available on the Internet at http://cnx.org/content/m0050/2.18 (May 2007).

7. H. KANG and T. SUNG, "An Analysis of Safety-Critical Digital Systems for Risk-Informed Analysis," *Reliab. Eng. Syst. Safety*, **78**, 307 (2002).

8. "Effects of Ethernet-Based, Non-Safety Related Controls on the Safe and Continued Operation of Nuclear Power Stations," Information Notice IN-2007-15, U.S. Nuclear Regulatory Commission (Apr. 17, 2007).

9. D. T. SMITH, T. A. DELONG, and B. W. JOHNSON, "A Safety Assessment Methodology for Complex Safety-Critical Hardware/Software Systems," *Proc. Int. Topl. Mtg. Nuclear Plant Instrumentation, Controls, and Human-Machine Interface Technologies*, Washington, D.C., November 13–16, 2000, American Nuclear Society (2000) (CD-ROM).

10. M. SINGHAL and N. G. SHIVARATRI, *Advanced Concepts in Operating Systems*, p. 297, McGraw-Hill Book Company, New York (1994).

11. R. S. PRESSMAN, *Software Engineering: A Practitioner's Approach, 6th Edition*, McGraw-Hill Book Company, New York (2005).

12. N. F. SCHNEIDEWIND and T. W. KELLER, "Applying Reliability Models to the Space Shuttle," *IEEE Software*, **9**, *4*, 28 (1992).

13. S. GUARRO, M. YAU, and M. MOTAMED, "Development of Tools for Safety Analysis of Control Software in Advanced Reactors," NUREG/CR-6465, U.S. Nuclear Regulatory Commission (1996).

14. J. DEVOOGHT and C. SMIDTS, "Probabilistic Reactor Dynamics—I: The Theory of Continuous Event Trees," *Nucl. Sci. Eng.*, **111**, 229 (1992).

15. T. ALDEMIR, "Utilization of the Cell-to-Cell Mapping Technique to Construct Markov Failure Models for Process Control Systems," *Proc. 1st Int. Conf. Probabilistic Safety Assessment and Management (PSAM 1)*, p. 1431, G. APOSTOLAKIS, Ed., Elsevier (1991).

16. T. ALDEMIR, "Quantifying Setpoint Drift Effects in the Failure Analysis of Process Control Systems," *Reliab. Eng. Syst. Safety*, **24**, 33 (1989).

17. P. C. CACCIABUE, A. AMENDOLA, and G. COJAZZI, "Dynamic Logical Analytical Methodology Versus Fault Tree: The Case Study of the Auxiliary Feedwater System of a Nuclear Power Plant," *Nucl. Technol.*, **74**, 195 (1986).

18. C. ACOSTA and N. SIU, "Dynamic Event Trees in Accident Sequence Analysis: Application to Steam Generator Tube Rupture," *Reliab. Eng. Syst. Safety*, **41**, 135 (1993).

19. M. HASSAN and T. ALDEMIR, "A Data Base Oriented Dynamic Methodology for the Failure Analysis of Closed Loop Control Systems in Process Plants," *Reliab. Eng. Syst. Safety*, **27**, 275 (1990).

20. C. ELKS, Y. Y. YU, M. REYNOLDS, and B. W. JOHNSON, "Quantitative Dependability Assessment of a Digital Feed Water Control System: Preliminary Results," UVA-CCS-QDA-001, Version 7, University of Virginia (2006).

21. B. LI, M. LI, and C. SMIDTS, "Integrating Software into PRA: A Test-Based Approach," C. SPITZER, U. SCHMOKER, and V. N. DANG, Eds., Springer-Verlag, London, United Kingdom (2004).

22. Y. ZANG and M. M. GOLAY, "Development of a Method for Quantifying the Reliability of Nuclear Safety-Related Software," *Proc. 6th Int. Conf. Probabilistic Safety Assessment and*

Management (PSAM 6), San Juan, Puerto Rico, June 20–23, 2002, Elsevier Science (2002) (CD-ROM).

23. M. MARSEGUERRA and E. ZIO, "Monte Carlo Approach to PSA for Dynamic Process Systems," *Reliab. Eng. Syst. Safety*, **52**, 227 (1996).

24. L. LAMPORT, S. SHOSTAK, and P. PEASE, "The Byzantine Generals Problem," *ACM Trans. Programming Languages and Systems*, **4**, 382 (1982).

25. J. KIRSCHENBAUM, M. STOVSKY, D. MANDELLI, P. BUCCI, T. ALDEMIR, D. W. MILLER, E. EKICI, and S. ARNDT, "A Benchmark System for the Assessment of Reliability Modeling Methods for Digital Instrumentation and Control Systems in Nuclear Plants," *Proc. 5th Int. Topl. Mtg. Nuclear Plant Instrumentation, Controls, and Human Machine Interface Technology (NPIC&HMIT 2006)*, Albuquerque, New Mexico, November 12–16, 2006, American Nuclear Society (2006).

26. M. HASSAN and T. ALDEMIR, "A Data Base Oriented Dynamic Methodology for the Failure Analysis of Closed Loop Control Systems in Process Plants," *Reliab. Eng. Syst. Safety*, **27**, 275 (1990).

27. B. LEIMKUEHLER and S. V. SHERIKAR, "Getting Optimum Performance Through Feedwater Control Valve Modifications," presented at 6th Electric Power Research Institute Symp. Valve Technology, Portland, Maine, July 14–16, 1997.

28. Y.-S. HU and M. MODARRES, "Evaluating System Behavior Through Dynamic Master Logic Diagram (DMLD) Modeling," *Reliab. Eng. Syst. Safety*, **64**, 241 (1999).

29. M. YAU, G. APOSTOLAKIS, and S. GUARRO, "The Use of Prime Implicants in Dependability Analysis of Software Controlled Systems," *Reliab. Eng. Syst. Safety*, **62**, 23 (1998).

30. T. ALDEMIR, M. P. STOVSKY, J. KIRSCHENBAUM, D. MANDELLI, P. BUCCI, L. A. MANGAN, D. W. MILLER, X. SUN, E. EKICI, S. GUARRO, M. YAU, B. W. JOHNSON, C. ELKS, and S. A. ARNDT, "Dynamic Reliability Modeling of Digital Instrumentation and Control Systems for Nuclear Reactor Probabilistic Risk Assessments," NUREG/CR-6942, U.S. Nuclear Regulatory Commission (2007).

31. S. GUARRO and G. EWELL, "Integrating MEMS Quality and Reliability Goals with the Use of Multi-Valued Logic Analysis," *Proc. 2nd Int. Conf. Integrated Micro/Nanotechnology for Space Applications (MNT99)*, Pasedena, California, April 11–15, 1999, Aerospace Corporation (1999).

32. M. YAU, M. MOTAMED, and S. GUARRO, "Nuclear Power Plant Digital System PRA Pilot Study with the Dynamic Flowgraph Methodology," *Proc. 5th Int. Topl. Mtg. Nuclear Plant Instrumentation, Controls, and Human Machine Interface Technology (NPIC&HMIT 2006)*, Albuquerque, New Mexico, November 12–16, 2006, American Nuclear Society (2006).

33. M. YAU, M. WETHERHOLT, and S. GUARRO, "Safety Analysis and Testing of Critical Space Systems Software," *Proc. 4th Int. Conf. Probabilistic Safety Assessment and Management (PSAM 4)*, New York, September 13–18, 1998, Springer Verlag (1999).

34. E. J. SHIELDS, G. APOSTOLAKIS, and S. B. GUARRO, "Determining the Prime Implicants for Multi-State Embedded Systems," G. APOSTOLAKIS and J. S. WU, Eds., *Proc. 2nd Int. Conf. Probabilistic Safety Assessment and Management (PSAM-II)*, San Diego, California, March 20–24, 1994, p. 7, International Association for Probabilistic Assessment and Management (1994).

35. K. RUSSELL, "Systems Analysis Programs for Hands-On Integrated Reliability Evaluations (SAPHIRE), Version 6.0: System Overview Manual," NUREG/CR-6532, U.S. Nuclear Regulatory Commission (1999).

36. "CAFTA for Windows, Version 3.0c," Science Applications International Corporation (1995).

37. "RISKMAN 7.1 for Windows," ABS Consulting (2003).

38. A. RAUZY, "New Algorithms for Fault Tree Analysis," *Reliab. Eng. Syst Safety*, **40**, 203 (1993).

39. T. ALDEMIR, "Computer-Assisted Markov Failure Modeling of Process Control Systems," *IEEE Trans. Reliab.*, **R-36**, 133 (1987).

40. L. DINCA and T. ALDEMIR, "Parameter Estimation Toward Fault Diagnosis in Nonlinear Systems Using a Markov Model of System Dynamics," *Nucl. Sci. Eng*, **127**, 199 (1997).

41. P. BUCCI, L. A. MANGAN, J. KIRSCHENBAUM, D. MANDELLI, T. ALDEMIR, and S. ARNDT, "Incorporation of Markov Reliability Models for Digital Instrumentation and Control Systems into Existing PRAs," *Proc. 5th Int. Topl. Mtg. Nuclear Plant Instrumentation, Controls, and Human Machine Interface Technology (NPIC&HMIT 2006)*, Albuquerque, New Mexico, November 12–16, 2006, American Nuclear Society (2006).

42. P. BUCCI, J. KIRSCHENBAUM, L. A. MANGAN, T. ALDEMIR, C. SMITH, and T. S. WOOD, "Construction of Event-Tree/Fault-Tree Models from a Markov Approach to Dynamic System Reliability," *Reliab. Eng. Syst. Safety*, **93**, *11*, 1616 (2008).

43. S. RUSSELL and P. NORVIG, *Artificial Intelligence: A Modern Approach*, Prentice-Hall, New Jersey (2003).

44. N. E. TODREAS and M. S. KAZIMI, *Nuclear Systems: Thermal Hydraulic Fundamentals*, Hemisphere Publishing Corporation (1990).

45. C. E. SHANNON and W. WEAVER, *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, Illinois (1964).