A Benchmark System for the Reliability Modeling of Digital Instrumentation and Control Systems

D. Mandelli[a1], T. Aldemir[a], J. Kirschenbaum[b], P. Bucci [b], D. W. Miller[a], M. Stovsky[b], E. Ekici[c], S. A. Arndt[d]

[a]The Ohio State University, Nuclear Engineering Program, Columbus, OH, USA
[b]The Ohio State University, Dept. of Computer Science & Engineering, Columbus, OH, USA
[c]The Ohio State University, Dept. of Electrical & Computer Engineering, Columbus, OH, USA
[d]U.S. NRC, Office of Nuclear Reactor Regulation, Washington, D.C. USA

Abstract: A candidate system is proposed as a benchmark for the assessment of methods for the reliability modeling of digital instrumentation and control (I&C) systems. The system under consideration is the digital feedwater control system (DFWCS) of a typical pressurized water reactor (PWR). A detailed description of the components, control laws, failure modes of the components and the communication logic is presented. A representation of the benchmark DFWCS as a finite state machine is also given. The finite state machine description is useful to: a) visualize and establish all the possible connections between the components (system topology), and, b) show how information regarding the status of the system is shared among system components. It is also shown how specific I&C failures, such as a failure in communications, are directly implemented in the model. Finally, requirements for such a benchmark system are revisited and discussed relative to the benchmark system presented.

Keywords: Dynamic PRA, Reliability modeling, I&C Reliability.

1. INTRODUCTION

In nuclear power plants, there is an accelerating trend to upgrade and replace analog instrumentation and control (I&C) systems with digital I&C systems. This transition from analog to digital I&C systems is due to the potential of digital I&C systems to improve reliability and safety of the plants [1]. As this replacement process continues, one or more methods addressing digital I&C system reliability are needed to quantify the change in the core damage frequency and large early release frequency of the plants by such systems [2].

A review of literature in this area appearing in NUREG/CR-6901 (Current State of Reliability Modeling Methodologies for Digital Systems and Their Acceptance Criteria for Nuclear Power Plant Assessments) [3] has identified several methods related to digital I&C system reliability modeling [4-19] and a characterized the requirements of such a method [20]. A conclusion of NUREG/CR-6901 was that there is no benchmark system available that can be used as the basis for an objective comparison of methods for digital I&C system reliability modeling. Recently, requirements for such a benchmark have been proposed based on the interactions within the digital I&C system and the interactions of the I&C system with the controlled/ monitored process [21, 22].

This paper presents the specifications for a benchmark system that meets the loosely control-coupled (LCC) system requirements shown in [21]. For LCC digital I&C systems, there is no direct dependency among the different system constituents, including software/ firmware. However, these systems may include dependencies through the controlled/monitored process. Section 2 describe in detailed the proposed benchmark including the description of the main components, the operating modes (Section 2.1), the control laws (Section 2.2) and the fault tolerant features of the system (Section 2.3). Section 3 shows how it is possible to model the logic of the benchmark system using a finite state machine.

[1] Mandelli Diego, mandelli.diego@gmail.com, The Ohio State University, Nuclear Engineering Program, 201 West 19th Avenue, Columbus, OH 43210

2. BENCHMARK SYSTEM

The benchmark system (see Fig.1) is based on the digital feedwater control system for an operating pressurized water reactor (PWR). The feedwater system serves two steam generators (SGs). 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 inches of the setpoint level (defined at 0 inches). The controller is regarded failed if water level in a SG rises above +30 or falls below -24 inches. 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 the flow of feedwater to the SGs to maintain a constant water level in the SGs

Each digital feedwater controller is comprised of several components (see Fig.1) which provide both control and fault tolerant capabilities. The control algorithms are executed on both a main computer (MC) and backup computer (BC). These computers produce output signals for the MFV, BFV, FP and pressure differential indicator (PDI) controllers. Each of these controllers forward the MC or BC's outputs to their respective controlled device (MFV, BFV or FP), or it can maintain the previous output to that device if the computers fail. When the controllers are used to maintain a previous output value to a controlled device in this way, it is necessary for operators to override the controller. The PDI controller enters into action in case of a loss of communication between the MFV and the MFV controller.

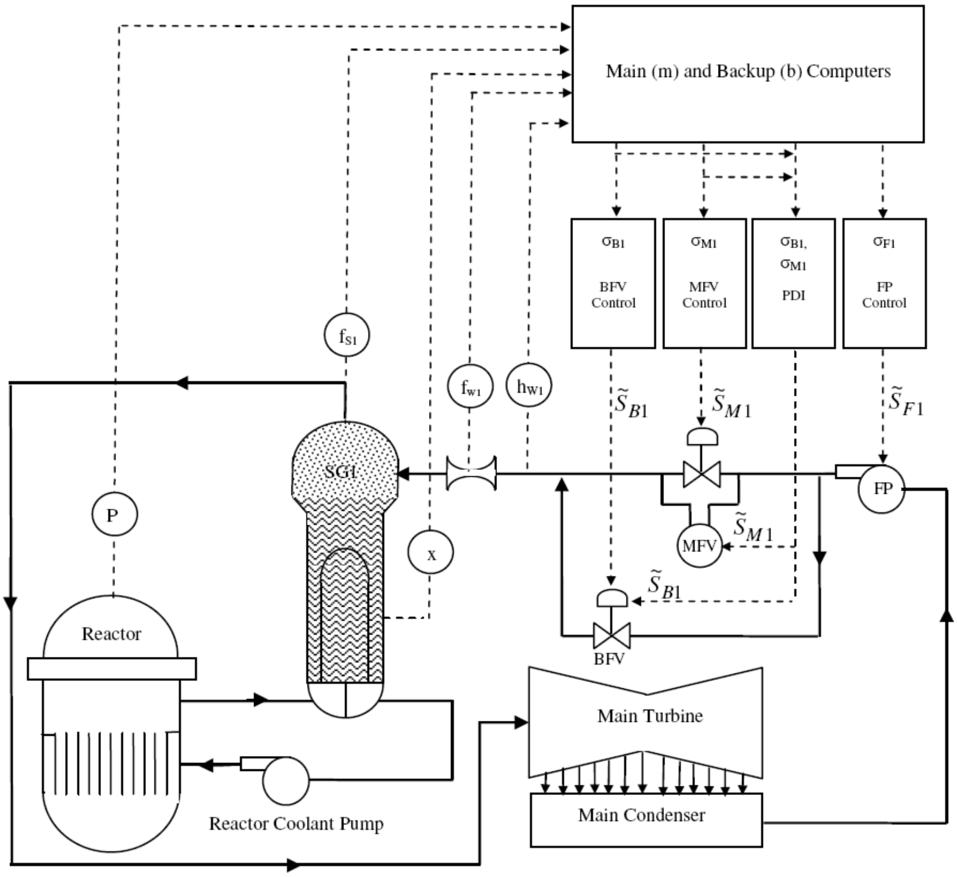


Fig.1 – Layout of the DFWCS

2.1 Operating modes

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

- Low power automatic mode
- High power automatic mode
- Automatic transfer from Low to High power mode
- 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.

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)

Transitions between Low and High power are controlled by the neutron flux readings. When the system is in Low Power mode and the neutron flux increases to a point when High Power mode is necessary, the MFV is signaled to open while the BFV is closed to maintain needed feedwater flow. The opposite transition occurs when the system is in High Power mode and the neutron flux decreases to a point when low power mode is needed.

2.2 Benchmark system control laws

In Low Power mode the control laws use the feedwater flow rate (f_{w1}), the steam flow rate (f_{s1}), feedwater temperature (h_{w1}), feedwater level in the steam generator (x_1), and neutron flux (P) to determine the BFV position. The feedwater level is fed to a proportional-integral (PI) controller algorithm using the feedwater temperature to determine the gain. Then this value is summed with the feedwater flow and neutron flux.

In High Power mode, the control laws again use f_{w1}, f_{s1}, h_{w1}, x_1 and P to compute the total feedwater demand. This computed value is used to determine both the position of the MFV and the speed of the FP. The feedwater flow and steam flow are summed and fed to a set of PI controller algorithms. The output from these controller algorithms is added to the feedwater level and that result is fed to a PI controller algorithm that uses the steam flow for the controller's gain.

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

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

Compensated Water Level:
$$\tau_2 \frac{dC_{Ln}}{dt} = -C_{Ln}(t) + x_n + \tau_1 \frac{df_{sn}}{dt} \tag{3}$$

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

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

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

FP Demand:
$$\sigma_{Fn}(t) = \begin{cases} \sigma_{Fn} & \text{If Low Power Operation} \\ \sigma_{Fn}(\max(C_{Fn}, \sigma_{Mn}^{-1}(C_{Fn})) & \text{If High Power Operation} \end{cases} \tag{7}$$

MFV Demand:
$$\sigma_{Mn}(t) = \begin{cases} \sigma_{Mn}(C_{Fn}) & \text{If High Power Operation} \\ 0 & \text{If Low Power Operation} \end{cases} \tag{8}$$

BFV Demand:
$$\sigma_{Bn}(t) = \begin{cases} 0 & \text{If High Power Operation} \\ C_{Bn}(t) & \text{If Low Power Operation} \end{cases} \tag{9}$$

$$\text{FP Speed:} \quad \tilde{S}_{Fn} = \begin{cases} \sigma_{Fnm} & MC \text{ Operational} \\ \sigma_{Fnb} & MC \text{ Failed, BC Operational} \\ \eta_{Fn} & MC \text{ Failed, BC Failed} \end{cases} \tag{10}$$

$$\text{MFV Position:} \quad \tilde{S}_{Mn} = \begin{cases} \sigma_{Mnm} & MC \text{ Operational} \\ \sigma_{Mnb} & MC\text{Failed, BC Operational} \\ \eta_{Mn} & MC \text{ Failed, BC Failed} \end{cases} \tag{11}$$

$$\text{BFV Position:} \quad \tilde{S}_{nB} = \begin{cases} \sigma_{Bnm} & MC \text{ Operational} \\ \sigma_{Bnb} & MC \text{ Failed, BC Operational} \\ \eta_{Bn} & MC \text{ Failed, BC Failed} \end{cases} \tag{12}$$

$$\text{PDI Decision:} \quad \tilde{S}_{Pn} = \begin{cases} 0 & \hat{S}_{Mn} > 0 \\ \eta_{Bn} & \text{Otherwise} \end{cases} \tag{13}$$

Also, 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. Equations 2 - 4 compute the flow demand for high power mode for the feedwater controller. Equation 6 computes the BFV demand for Low power mode. The dynamic gains $\beta_{Bn}(h_{wn})$ and $\lambda_{Mn}(\sigma_{Mn})$ in Eq. 6 are obtained from lookup tables on the feedwater temperature and the MFV opening respectively. The subscripts *m* and *b* in Eqs. 10-13 refer to signals from the main and backup CPUs respectively. The η_{Fn} , η_{Mn} and η_{Bn} in Eqs. 10-13 denote history data for the FP, MFV and BFV positions, respectively. If both the MC and the BC have failed, these data are used to determine the FP, MFV and BFV positions.

2.3 Fault Tolerant Features of Benchmark System

Since the MFV, BFV and FP controllers forward the control signals to the corresponding control points (the MFV, BFV, and FP, respectively, as well as the PDI controller), they provide a level of fault tolerance if both the MC and BC fail by allowing the operators time to intervene by holding the outputs of each to a previously valid value.

The MC and BC, the MFV, BFV and FP and the PDI controllers are each connected to an independent power source wired to a separate bus. A single power source failure can only affect 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 3 times the Nyquist criterion to avoid aliasing. Moreover, a failure in the MC or BC can be detected and the fail over [2] to a healthy component can occur with enough time to meet the response requirements of the process.

The water level set point is taken from a switch connected to the MFV and is propagated to both the MC and BC. If the set point signal goes out of range, then the computers fall back on a preprogrammed set point value.

Each computer (MC or BC) is connected to a watchdog timer. A watchdog timer is a hardware timer and associated connections used to determine if a software error or other computer failure has rendered a processor unusable. A normally functioning computer resets the watchdog timer at regular, defined intervals so the timer does not “go off.” However, in the presence of a software error or another computer failure, the timer will not be reset by the computer and the timer can go off. For example, a runaway process, halted (failed) processor, or a sufficiently lengthy computational delay may result in failure to reset the watchdog timer. As a result, the watchdog timer may go off. If the timer goes off, all components in the controller connected to the watchdog timer are notified of the computer failure. In the case of the benchmark system, the MFV, BFV, and FP controllers are notified and transfer control away from the affected computer.

Each computer (MC or BC) 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

[2] Fail over is the process in which a degraded component is removed from control and replaced by a healthy component

to digital conversion of the values. Each computer will ignore input that fails these checks if the other inputs are still valid.

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 the other computer.

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

The MFV, BFV and FP controllers also send their outputs to the MC and BC. When the MC (or BC) is in control, it compares its output to the signals that the MFV, BFV and FP controllers output signal to the actuators. If the output signal differs, then the computer indicates to the MFV, BFV and FP controllers that it has failed. 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 value from the computer (essentially the last one that the controller can store) and recycle that value to the control points. Any time a component fails, the operator console is notified to allow operators to take mitigating actions.

3. FINITE STATE MACHINE MODELING

The design of a system finite state machine for digital control systems takes into account several aspects that can be summarized as follows:

- *The structure and the topology of the network*. The concept of topology applied to networks refers to how data are shared and passed among the components connected to the network itself.
- *The format of the data.* From a reliability view point, the format of the data affects the resilience property of the communication systems from external events and how the communication system itself is able to detect and correct errors in the data transmitted on the line.
- *Transmission media*. Different types of media can be used to transmit data. Guided (e.g., coaxial or fiber-optic cable) and unguided (e.g., electromagnetic waves used without a physical conductor) media can be used.

The input for the finite state machine modeling is the failure modes and effects analysis (FMEA) of the DFWCS (see Table 1). These failure classes may include sensor failures, output failures, input failures and internal failures. Each of the failure classes may contain a large number of faults. For example, sensor failure may be the result of a physical sensor failure, cut wires, loose connections, or hardware (such as analog to digital converters) on the receiver failing. While these failure classes may be general, they are expected to capture the necessary information about possible failures of the benchmark feedwater control system.

Each component type in the system, MC, BC, MFV controller, BFV controller, FP controller and PDI controller has a separate FMEA chart associated with that component. In addition, the actuated devices (i.e. MFV, BFV and FP) may fail to perform their design due to mechanical failure. The only mechanical failures that will be considered for the benchmark DFWCS are the valves getting stuck in their current position.

From the FMEA showed in Table 1, the DFWCS can be regarded as consisting of three layers of interaction:

- Intra-computer interactions: a layer which describes the status of the single computer (MC or BC)
- Inter-computer interactions: a layer which describes the status of the set of both computers (MC and BC)
- Computer-controller-actuated device interactions: a layer which describes the status of the controllers (MFV, BFV and FP controllers).

In Sections 3.1, 3.2 and 3.3 these three layers are presented and described.

Table 1 –FMEA for the DFWCS components

Component	Failure Mode	Effect on the System
Sensor	Out of range	MC and BC detect this failure and ignore the input from the sensor..
	Loss of output	No input to MC and BC.
MFV	Stuck	MFV maintains its position.
MFV Controller	Loss of input	Controller initiates failover. If the failover fails, the valves and FP remain in the same state.
	Loss of output	If detected by PDI, old signal for the MFV is used. If undetected, the valves and FP remain in the same state.
	Operating but not able to detect failures	The MFV controller will not detect any computer failures.
	Down	MFV controller may output any signal to the MFV.
BFV	Stuck	BFV maintains its position.
BFV Controller	Loss of input	Controller initiates failover. If the failover fails, the valves and FP remain in the same state.
	Loss of output	If detected by PDI, old signal for the BFV is used. If undetected, the valves and FP remain in the same state.
	Operating but not able to detect failures	The BFV controller will not detect any computer failures.
	Down	BFV controller may output any signal to the BFV.
FP	Stuck	FP maintains its speed.
FP Controller	Loss of input	Controller initiates failover. If the failover fails, the valves and FP remain in the same state.
	Loss of output	If detected by PDI, old signal for the FP is used. If undetected, the valves and FP remain in the same state.
	Operating but not able to detect failures	The FP controller will not detect any computer failures.
	Down	FP controller may output any signal to the FP.
PDI Controller	Loss of input	The PDI will be unable to detect MFV controller failures.
	Loss of output	The PDI will be unable to mitigate MFV controller failures.
	Down	PDI controller may output any signal to the MFV.
MC	Loss of 1 input	The MC waits for 1 cycle before entering the Down state.
	Loss of 2 inputs	The MC waits for 1 cycle before entering the Down state.
	Intermittent failure	Control is transferred to the BC. When the MC becomes operational again, it acts as a backup.
	Down	MC indicates its failure to the MFV, BFV and FP controllers. If control is not transferred to the BC, arbitrary output can be produced by the MC.
BC	Loss of 1 input	The BC waits for 1 cycle before entering the Down state.
	Loss of 2 inputs	The BC waits for 1 cycle before entering the Down state.
	Intermittent failure	Control is transferred to the MC. When the BC becomes operational again, it acts as a backup.
	Down	BC indicates its failure to the MFV, BFV and FP controllers. If control is not transferred to the MC, arbitrary output can be produced by the BC.

3.1. Intra-computer interactions

The intra-computer interactions layer consists of 5 states (see Fig.2). These interactions can be regarded as transitions between the possible states of a single computer. In State A, the computer is operating correctly. In State B, the computer detects loss/invalid output for 1 sensor of any type (e.g. water level). State C represents loss/invalid output for 2 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 is transmitting the wrong information to the controllers. These states capture the possible failures in the FMEA table presented Table 1.

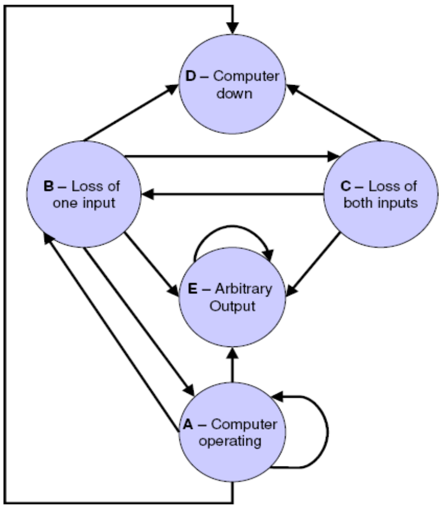


Fig.2 –Intra-computer interactions

3.2. Inter-computer interactions

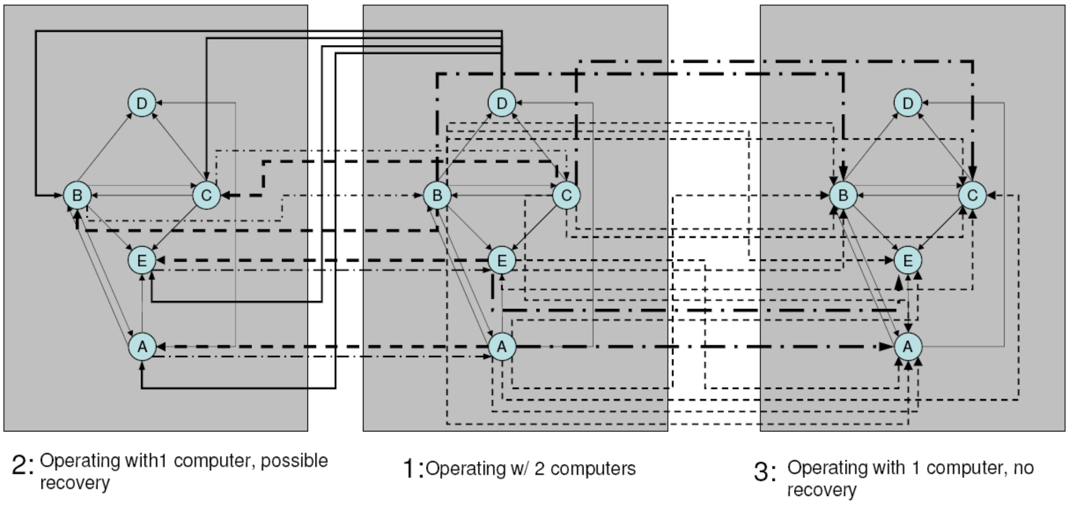
The inter-computer interaction layer displayed in Fig.3, shows the interactions between the two computers (MC and BC). In particular, the transfer of control from the MC to the BC is represented here. In this layer, 3 computer macro-states (MSs) are identified. Each of these macro-states indicates the status of both the computers:

- In State 1 both MC and BC are operating normally.
- In State 2, one computer is operating correctly and the other is down but can be recovered.
- In State 3, again one computer is operating correctly and the other is down but it is not recoverable.

Transitions between the MSs depend upon the state of the controlling computer. 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. Both the MC and BC can be the primary or the secondary computer.

Transitions from MS1 to MS2 are due to recoverable failures while transitions from MS1 to MS3 are due to not recoverable failures. Recoverable and non-recoverable failures are defined as the following:

- Recoverable failure corresponds to the momentarily 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 of the sensors). Since, it is possible to recover this failure: transitions from MS2 to MS1 are possible.
- Non-recoverable 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. Since it is not possible to recover this failure: transitions from MS3 to MS1 are possible.


D
B
C
E
A
D
B
C
E
A
D
B
C
E
A
2: Operating with1 computer, possible recovery
1 :Operating w/ 2 computers
3: Operating with 1 computer, no recovery

Computer States

A: Operating **B:** Loss of one input
C: Loss of both inputs **D:** Computer down
E: Arbitrary output

Macro States

1: Controller is receiving data from both computers
2: Controller is receiving data from 1 computer while the other one can be recovered
3: Controller is receiving data from 1 computer while the other one can not be recovered
Freeze: Controller sends the same data to the valves from the previous time step

— — — ·	Secondary goes down (recoverable)	———	Primary release control of the process.	— · — ·	Secondary computer watchdog timer trips or loss of output to controller	- - - - - - -	Common cause sensor failure
- · - · - · -	Secondary recovers	- - - - - - -	Primary computer watchdog timer trips or loss of output to controller.	— — ·	Primary goes down. Secondary unavailable		

Fig.3 – Inter-computers interactions

3.3. Computer-controller-actuated device interactions

Figure 4 shows all the possible controller-computer-actuated device interactions according to the FMEA chart presented in Table 1. The shaded circles represent signals to the actuated devices (MFV, BFV, FP) upon computer/controller failure, as well as the mechanical failure of the actuated device (Device Stuck). As indicated in the Table 1, mechanical failure of the actuated device leads to the device maintaining its current position for MFV and BFV or to zero flow for FP.

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 controller and actuated device are restored.

The scheme shown in Fig.4 shows the connection between a single controller (e.g., the MFV controller) and the computers (MC and the BC) and its own actuated device (MFV). In particular, the following types of controller failures are under consideration:

- Arbitrary output: random data are generated and sent to the actuated device (i.e., MFV, BFV and FP)
- Output high: output value is stuck at the maximum value (i.e. valve totally open or pump at the maximum speed)
- Output low: output value is stuck at the minimum value (i.e. valve totally closed or pump stopped)
- 0 vdc output: loss of communications between controller and actuated device
- mechanical failure of the actuated device

Moreover, as a result of the failure of both computers, the controller can recognize the failure and send to the actuated devices (pump or valves) the old valid value (Freeze). If the controller does not recognize the failure, then it will simply pass on false information (Arbitrary Output) to the actuated device. Figure 4 also shows how the computer-computer interactions (presented in Fig.3) integrate with computer-controller and controller-actuated device interactions.

Device Stuck refers to mechanical failures and is independent of the failure modes of the computers and controllers. The behavior of the controller under normal and failed operation can be described as follows:

- 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.
- If the controller is operating and an Output High or an Output Low or an 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.
- 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 actuated device assumes the highest, the lowest or an arbitrary position, respectively.
- If a loss of output occurs when the controller is failed (i.e. the controller is sending Arbitrary Output, Output High or Output Low state), then the actuated device receives a 0 vdc as input which correspond to the lowest position.

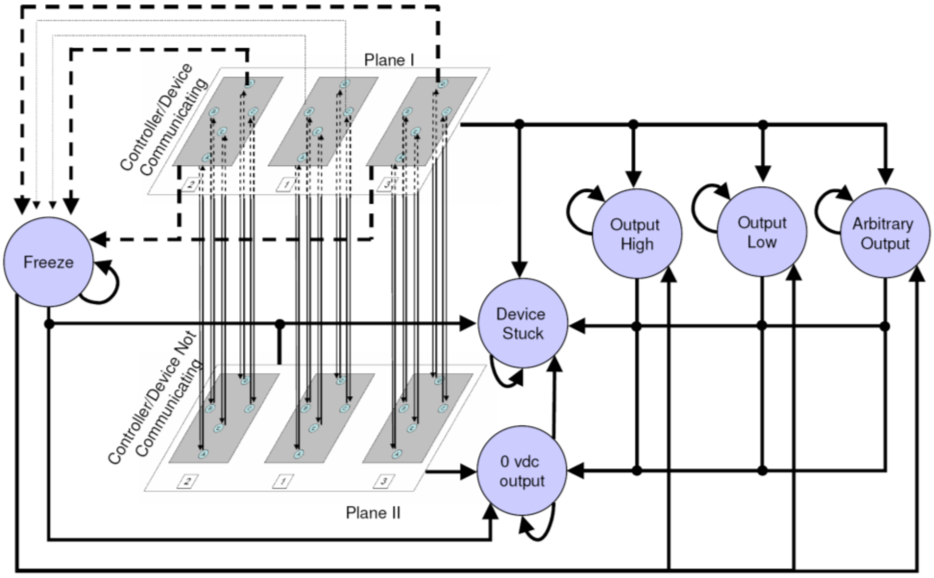


Fig.4 – Computer-controller-actuated device interactions

4. CONCLUSION

The DFWCS similar to that of an operating PWR is proposed as a benchmark system for the comparison of methodologies for the reliability modelling of digital I&C systems. The benchmark system incorporates most of the features that are representative of LCC digital I&C systems and can be used as the basis for an objective comparison of methods for digital I&C system reliability modelling. The digital features included in the benchmark are a clock that drives progress, the representation of power requirements, real time constraints on processing, the use of polling and interrupts, self-diagnostics including the use of a watchdog timer, and interactions with the controlled/monitored process. These features ensure that the benchmark system is sufficient to represent the possible deployments of an LCC system.

Acknowledgements

The research presented in this paper was sponsored by the U.S. Nuclear Regulatory Commission (US NRC). The information and conclusions presented here in are those of the authors and do not necessary represent the views or positions of the US NRC. Neither the U.S. Government nor any

agency thereof, nor any employee, makes any warranty, expressed or implied, or assume any legal liability or responsibility for any third party's use of this information.

References

[1] "Guideline for Performing Defense-in-Depth and Diversity Assessments for Digital I&C Upgrades - Applying Risk-Informed and Deterministic Methods", 1002835, EPRI, Palo Alto, CA (2004)

[2] "Advisory Committee on Reactor Safeguards", Regulatory Guidance on Implementation of Digital I&C Systems (1997).

[3] T. Aldemir, D. W. Miller, M. Stovsky, J. Kirschenbaum, P. Bucci, A. W. Fentiman, and L. A. Mangan, "Current State of Reliability Modelling Methodologies for Digital Systems and Their Acceptance Criteria for Nuclear Power Plant Assessments", NUREG/CR-6901, U. S. Nuclear Regulatory Commission, Washington, D.C. (2006)

[4] C. J. Garret and G. E. Apostolakis, "Automated Hazard Analysis of Digital Control Systems", Reliab.Engng & System Safety, 77, 1-17 (2002).

[5] 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, Washington, D.C. (1996)

[6] D. T. Smith, T. A. Delong and B. W. Johnson, "A Safety Assessment Methodology for Complex Safety-Critical Hardware/Software Systems", International Topical Meeting on Nuclear Plant Instrumentation, Controls, and Human-Machine Interface Technologies, Washington, D.C. (2000).

[7] N. F. Scheneidewind and T. W. Keller, "Applying Reliability Models to the Space Shuttle", IEEE Software, 28-33 (1992).

[8] Y. Zang and M. M. Golay, "Development of a Method for Quantifying The Reliability of Nuclear Safety-Related Software", PSAM6: Proceedings of the 6th International Conference on Probabilistic Safety Assessment and Management, CD-ROM Version, Elsevier Science Ltd. (2002).

[9] G. Pai, S. Donohue and J. Dugan, "Estimating Software Reliability From Process and Product Evidence", Elsevier Science Ltd. (2002).

[10] P. L. Goddard, "A Combined Analysis Approach to Assessing Requirements for Safety Critical Real-Time Control Systems", IEEE Proceedings Annual Reliability Maintainability Symposium, Hughes Aircraft Company (1996).

[11] A. Raunzy, "Mode Automata and Their Compilation into Fault Trees", Reliab.Engng & System Safety, 78, 1-12 (2002).

[12] Balakrishman, M. and K. Trivedi, "Stochastic Petri Nets for Reliability Analysis of Communication Network Applications With Alternate Routing", Reliab.Engng & System Safety, 53, 243-259 (1996).

[13] J. L. Peterson, "Petri Nets", ACM Computing Surveys, 9 (1977).

[14] M. Marsan and G. Conte, "A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems", ACM Transactions on Computer Systems, 2, 93-122 (1984).

[15] T. S. Liu and S. B. Chiou, "The Application of Petri Nets to Failure Analysis", Reliability Engineering and System Safety, 129-142 (1997).

[16] B. Li, M. Li and C. Smidts, "Integrating Software into PRA: A Test-Based Approach", Springer – Verlag, London, U.K. (2004).

[17] C. Smidts and M. Li, Validation of A Methodology For Assessing Software Quality, UMD_RE_2002-07, University of Maryland, College Park, MD, (2-1-2002)

[18] N. F. Scheneidewind, "Analysis of Error Processes in Computer Software", Proc.Int'l Conf.Reliable Software, 76-78, IEEE CS Press (1975).

[19] L. M. Kaufman and B. W. Johnson, Embedded Digital System Reliability and Safety Analyses, NUREG/GR-0020, U.S. Nuclear Regulatory Commission, Washington, D.C. (2001)

[20] S. A. Arndt, E. Thornsbury and N. O. Siu, "What PRA Needs From a Digital System Analysis", E. J. Bonano, A. L. Camp, M. J. Majors and R. A. Thompson (Eds.), Probabilistic Safety Assessment and Management, Elsevier Science Publishing Co., New York (2001).

[21] J. Kirschenbaum, M. Stovsky, P. Bucci, T. Aldmir and S. A. Arndt, "Benchmark Development for Comparing Digital Instrumentation and Control System Reliability Modeling Approaches", Proceedings of American Nuclear Society International Topical Meeting on Probabilistic Safety Assessment, (2005).

[22] T. Aldemir, D.W, Miller, M. Stovsky, J. Kirschenbaum, P. Bucci, L.A. Mangan, A. Fentiman and S.A. Arndt, “Methodologies for the Probabilistic Risk Assessment of Digital Reactor Protection and Control Systems,” Nuclear Technology, Vol. 159, No. 2, pp 167-191, August, 2007