

# EVALUATION OF I&C ARCHITECTURE ALTERNATIVES REQUIRED FOR THE JUPITER ICY MOONS ORBITER (JIMO) REACTOR

Michael D. Muhlheim,<sup>1</sup> Richard T. Wood,<sup>1</sup> William L. Bryan,<sup>1</sup> Thomas L. Wilson, Jr.,<sup>1</sup>  
David E. Holcomb,<sup>1</sup> Kofi Korsah,<sup>1</sup> and Usha Jagadish<sup>2</sup>

<sup>1</sup>Oak Ridge National Laboratory

1 Bethel Valley Road, Oak Ridge, Tennessee 37831

muhlheimd@ornl.gov

<sup>2</sup>Athma-Tech, Inc.

306 Treyburn Drive, Knoxville, Tennessee 37934

**Abstract** — *This paper discusses alternative architectural considerations for instrumentation and control (I&C) systems in high-reliability applications to support remote, autonomous, inaccessible nuclear reactors, such as a space nuclear power plant (SNPP) for mission electrical power and space exploration propulsion. This work supported the pre-conceptual design of the reactor control system for the Jupiter Icy Moons Orbiter (JIMO) mission. Long-term continuous operation without intermediate maintenance cycles forces consideration of alternatives to commonly used active, N-multiple redundancy techniques for high-availability systems.*

*Long space missions, where mission duration can exceed the 50% reliability limit of constituent components, can make active, N-multiple redundant systems less reliable than simplex systems. To extend a control system lifetime beyond the 50% reliability limits requires incorporation of passive redundancy of functions. Time-dependent availability requirements must be factored into the use of combinations of active and passive redundancy techniques for different mission phases. Over the course of a 12 to 20-year mission, reactor control, power conversion, and thermal management system components may fail, and the I&C system must react and adjust to accommodate these failures and protect nonfailed components to continue the mission. This requires architectural considerations to accommodate partial system failures and to adapt to multiple control schemes according to the state of nonfailed components without going through a complete shutdown and restart cycle.*

*Relevant SNPP I&C architecture examples provide insights into real-time fault tolerance and long-term reliability and availability beyond time periods normally associated with terrestrial power reactor I&C systems operating cycles. I&C architectures from aerospace systems provide examples of highly reliable and available control systems associated with short- and long-term space system operations.*

*Reliability concepts are discussed, and differences between various redundancy management schemes are compared. Mission time-dependent availability requirements indicate that a SNPP I&C might employ different types of redundancy at different times in a mission. Conclusions are drawn regarding appropriate architectural features relative to mission duration and control system availability requirements.*

## I. INTRODUCTION

The National Aeronautics and Space Administration (NASA) is currently considering deep space missions that would utilize a space nuclear power plant (SNPP) to provide energy for propulsion and spacecraft power. A SNPP provides the opportunity to supply high-sustained power for space applications that is both reliable and mass efficient.

SNPP instrumentation and control (I&C) system architectures for a long-term, unattended, and autonomous reactor controller must enable continuous operation of the SNPP without shutting down for a 12 to 20-year mission. For this study, a baseline architecture for a long mission

duration was analyzed, and alternative I&C architectures from aerospace systems, which serve as examples of highly reliable and available control systems, were reviewed.

## II. OPERATING REQUIREMENTS

The SNPP is intended to power a spacecraft that was to explore the three icy moons of Jupiter. Because of the large distances between the Jupiter Icy Moons Orbiter (JIMO) spacecraft and Earth, an autonomous reactor control system needs to have the authority to make decisions for operation of the reactor. This is necessary for deep space missions because the required reaction

time of the reactor I&C system will be much shorter than the communications time from ground controllers for most of the mission. In addition, component maintenance is not feasible during the mission, so the reactor control system must also employ some form of component redundancy and fault management to accommodate fault detection and recovery. To summarize, the I&C system must be able to control the reactor, respond to design basis faults, provide automatic system recovery and reconfiguration<sup>1</sup> given a reactor module or power conversion module fault, all with minimal intervention from ground controllers.

The concept of operations for the JIMO SNPP is to bring the reactor and two of three or four redundant Brayton power converters up to full power, after which the reactor and Brayton converters would operate at power for 12 to 20-years without being shutdown.

Unlike terrestrial-based reactor controllers, the SNPP I&C system must be able to maintain a very high reliability for 12 to 20-years without maintenance. This requires architectural considerations to accommodate partial system failures and to adapt to multiple control schemes according to the state of nonfailed components without going through a complete shutdown and restart cycle.

After reactor and power converter system start-up and load balancing, reactor control will consist of a relatively slow control loop operation that will periodically and purposely change the position of every control rod or reflector. This is necessary to prevent sticking of these component movements and to gradually adjust their nominal positions to compensate for reactor fuel burnup. During this nearly steady state of reactor operation, a temporary loss of reactor control could be tolerated while reactor controller fault recovery and reinitialization operation occurs, if necessary. This is similar to other long-term space mission controllers where temporary loss of control can be tolerated as long as control from primary controllers or backup units in a standby redundant controller configuration can eventually be regained through fault recovery processes.

Unlike controllers used in long-term space missions, a temporary loss of control of the SNPP during a transient

or upset condition cannot be tolerated. If there is a failure in any of the Brayton units or other subsystems, the SNPP control system must take immediate action, without a total shutdown of the reactor and/or the Brayton units, to rebalance the total plant thermal power to prevent damage to the reactor or the remaining Brayton converters. Since the SNPP is the sole source of electrical power to maintain spacecraft and power system operations, it must be continuously available to supply power.

Surviving an upset, faulty signal, or equipment failure requires the controllers to have parallel redundancy with fault masking or hot sparing with immediate switchover backup operation. Under conditions of partial system failure, temporary suspension of control could result in damage to the reactor or a Brayton unit, or an unrecoverable shutdown. Without a backup electrical supply, there would be no way to restart the reactor and power converters to continue the mission in any capacity. However, with a hot run back, fault recovery scheme, the mission could possibly continue under limited capacity or the initiating faults could be diagnosed and perhaps recovered to full capacity.

### III. OTHER ARCHITECTURES

#### III.A. Spacecraft Avionics and Controls Architecture

Spacecraft avionics provide examples of high-reliability, high-availability control systems. There are two general categories of spacecraft avionics systems—those for long mission spacecraft, such as communications satellites or interplanetary exploration spacecraft, and those associated with flight and launch systems. Time criticality of control and man-rated safety requirements generally distinguish these two types of control systems and their architectures.

Long mission avionics and control architecture goals are to be available over extremely long timeframes and to be able to recover from anticipated but unpredictable upsets or faults. Control strategies for long missions have to include autonomous reconfiguration decision capability and must be able to tolerate periods of no control while faulted systems recover. Eventual fault recovery is the design goal. For the limited periods of time when continuous control is absolutely required during a long mission, the control architecture must configure itself to operate in a redundant, fault-masking configuration.

Launch system avionics, and manned-flight launch systems in particular, have very stringent availability and fault tolerance requirements that are comparable to a reactor protection system in a commercial nuclear power

---

<sup>1</sup> A change in the system's configuration in response to some triggering event is defined as a reconfiguration. A reconfiguration occurs when the system is reinitialized because of a logical subsystem failure or when the system degrades to a lesser number of subsystems or a less redundant subsystem because no spares exist to replace a failed component.

plant. Most of these systems require guaranteed correct operation in the presence of one or more temporary or even permanent system component failures. These systems are normally short-term missions and are engineered and maintained to operate well within system component wear-out times. System faults would normally occur because of environmental over-stress or temporary upset due to radiation or electrical transients. Modular redundancy and design diversity are used to guard against common-mode design errors.

### III.B. Space Shuttle Avionics Architecture

The Space Shuttle avionics control system and architecture is fairly unique, relatively complicated and very flexible because the spacecraft is a man-rated launch system, an on-orbit experiment platform, and a man-rated reentry system with atmospheric flight and landing. In addition, it is a reusable system that is serviced and put through the launch, orbit, reentry, and landing cycle many times. The Space Shuttle is designed for fully remote and automated operation as well as manual, pilot-operated flight and maneuvering. The Space Shuttle design was one of the earliest fly-by-wire avionics control systems and one of the earliest examples of a reconfigurable, modular redundant control system.

The Space Shuttle operation involves four distinct operational phases: prelaunch readiness checking; vertical launch to low Earth orbit; on-orbit flight and experiments; Earth atmosphere reentry, zero power atmospheric flight maneuvering, and wheels down, horizontal landing. These different operational phases have different control system reliability and availability requirements due to differences in the criticality or speed of response for the Shuttle's control during a particular phase.

The Space Shuttle avionics system (referred to as the Digital Processing System or DPS) consists of five general-purpose computers (GPCs) that are interconnected with each other, other sensor and actuator subsystems, and I/O devices via 24 serial data busses. Discrete sensor and actuator I/O and proportional sensors and actuators are handled by multiplexer/demultiplexer processing units distributed throughout the Space Shuttle, external fuel tank, and the two solid rocket boosters. The network of serial busses ensures all GPCs, I/O processors, master timing units, communications units and ground support interfaces are uniformly connected. In addition to the serial busses, dedicated point-to-point discrete I/O lines interconnect the five GPCs to provide tightly coupled time synchronization during modular redundant operations. Through this interconnection scheme, there is no distinction in any GPC's function. Any GPC can

assume any particular function in any of the different mission phase configurations simply by running different software.

Most sensors, discrete I/O processors and actuators are configured with multiple redundant components. The multiple serial busses allow redundant sensor or actuator channels to be used by every processor in the quintuple modular redundant (QMR) configurations and control calculations.

### III.C. Unmanned Spacecraft Avionics Development

The history of long mission spacecraft controllers spans from the time before large-scale integrated electronic components to the present where multiple CPU cores are available. Long-duration spacecraft controllers have been engineered for long-term survivability through quality engineering of components, self-testing hardware and software, fault recovery techniques, multiple reconfigurable redundant warm and cold spare units and mission contingency plans for gradual degradation of operations as system component failures occur. The other feature that must be designed into interplanetary spacecraft is autonomous control and fault recovery. As the spacecraft travels to the outer reaches of the solar system, speed of light transit time and lack of radio contact due to solar or other eclipsing bodies limits the ability to perform any sort of remote fault detection or recovery actions between Earth stations and a probe. Goals, directives, or software updates can be transmitted to the spacecraft, but fault recovery and real-time control are essentially autonomous.

Unmanned spacecraft controllers do not have to handle spacecraft launch control and only have to handle landing control if the spacecraft involves a lander with rocket-assisted deceleration and maneuvering. Unmanned spacecraft do control in-space maneuvering rockets, but with less criticality than launch or landing maneuvering due to the path length of trajectories and the allowable time for additional fine course correction. If a temporary fault or upset in some part of the spacecraft controller causes the spacecraft to miss a scheduled maneuver, the mission can continue if the controller fault can be detected, reset, and the controller returned to operation within some reasonable time with a corrective maneuver operation performed after a faulted controller is recovered.

Current state-of-the-art for control of long-duration spacecraft involves multicluster distributed computing throughout the spacecraft. Control is distributed according to functional divisions of subsystem control,

data management, and communications. Clusters of microprocessors are employed to provide robust fault tolerance and recovery through reconfigurable warm and cold redundant spares. Redundant interconnect busses with access control guardian circuitry are employed to isolate subsystem faults from intersubsystem communications links. Temporary or permanent electronic failures are considered to be primary fault sources with wiring and interconnect failures considered to be rarer than wiring design errors. The use of distributed computer control improves the spacecraft's ability to continue a mission given a failed subsystem or experiment without impacting any other subsystem or experiment. In some cases, architectures have been proposed that would allow subsystems with failed computer clusters to "borrow" a computer out of another subsystem's cluster to continue some or all of its mission.

#### IV. MEASURES OF RELIABILITY

Reliability is a measure of the likelihood that the system has not experienced any failures. Both qualitative and quantitative reliability analyses can provide insights into the reliability, fault tolerance, diversity, and redundancy of alternative design options. Qualitatively, the number of component failures that can cause system failure can provide an idea of the importance of that failure mode. As the number of component failures required for system failure increases, the likelihood of that combination of failures causing system failure decreases. Quantitatively, different architectures can be compared even with minimal data and first-of-a-kind components by setting component failure rates in relation to each other.

Fault-tolerant designs are important for computer and communication systems used in satellite control where online/onboard manual intervention to repair or replace a failed component is difficult or impossible. The inability to perform maintenance (i.e., the system behaves as a nonrepairable system) demands that satellite systems not only be single-fault tolerant, but be capable of withstanding multiple faults and to be able to manage the faults as they occur. Thus, automatic recovery and reconfiguration mechanisms play a crucial role in implementing fault tolerance because an uncovered fault may lead to a system failure even when adequate redundancy exists. This happens if a faulty unit that is not reconfigured out of the system produces incorrect results that contaminate nonfaulty units [1].

Redundancy alone does not guarantee fault tolerance. The only thing redundancy guarantees is a higher fault arrival rate compared to a simplex system of the same

functionality. For a redundant system to continue correct operation in the presence of a fault, the redundancy must be properly managed [2]. Redundancy management issues are deeply interrelated and determine not only the ultimate system reliability but also the performance penalty paid for fault tolerance.

Four methods of fault protection are used to achieve a high reliability for systems:

1. design and implementation standards where the reliability of every constituent component is as high as reasonably achievable,
2. system-level modular redundancy with functionally equivalent elements executing identical tasks in parallel where results of individual redundant subsystems can be evaluated against a majority to prevent propagation of individual subsystem errors from control operations,
3. design diversity of duplicate but independently developed functions to guard against generic design errors in an I&C controller of duplex or N-modular designs, and
4. added hardware and/or software for fault detection and recovery where self-testing is constantly monitoring for generic hardware or software failures and signaling redundant subsystems or supervisory systems to reset and restart failed functions or to logically reconfigure the control system around the failed subsystem.

Software can also be used to implement fault tolerance against hardware faults by

1. fault detection, such as the software voting on results of replicated processors;
2. fault isolation, where the software executes self-testing programs;
3. repair by switching off the failed subsystem; and
4. recovery by reinitializing a failed task.

Hardware redundancy can be implemented in static, dynamic, or hybrid configurations [2]. Static (or passive) redundancy techniques do not detect or explicitly perform any reactive action to control errors, but rather rely on masking to simply prevent error propagation beyond predefined error containment boundaries.

Static redundant systems are generally implemented for short-term, high-availability missions such as the Space Shuttle, commercial airliners, or a reactor safety system where the control system must continue to perform its intended function in the presence of temporary or even permanent failure of some components. Failures

or upsets can be caused by temporary electrical upsets, actual part wear-out, or loss of timeframe or synchronicity. For static redundant systems, failure within a subset of the redundant controller is masked through some control output consensus assurance mechanism. A failed component can be diagnosed and replaced at a noncritical operation time. The highest reliability for static redundant systems occurs when the mission length is significantly less than the wear-out time of any component in the system. When a mission time exceeds the 50% reliability lifetime for any constituent part of a static redundant system, the overall system reliability actually decreases more rapidly than in a simplex system of the same functionality.

Dynamic (or active) redundancy techniques use fault detection followed by diagnosis and reconfiguration. Logical subsystems can be reconfigured [3]. Before component failures cause systems to irretrievably fail, the system can recover by logically replacing failed components with spares. If insufficient spares are available, the system can degrade to a lesser number of subsystems or a less redundant subsystem. When a logical subsystem fails, the system also fails unless it can be reinitialized by a separate subsystem or component. Spares may also have N-modular redundancy (NMR) and may or may not be active. Masking (typically by voting) is not used in dynamic redundancy; instead errors are handled by actively diagnosing error propagation and isolating or logically replacing a faulty component.

Dynamic or reconfigurable redundant systems are used for missions that approach or exceed wear-out times of constituent components in the control system, such as earth satellites or outer planet exploration missions. In these long-term redundant systems, electrical wear-out is calculated based on energized time or radiation exposure time. Redundant spare components or subsystems are usually turned off electrically until needed to keep the wear-out clock at zero until needed.

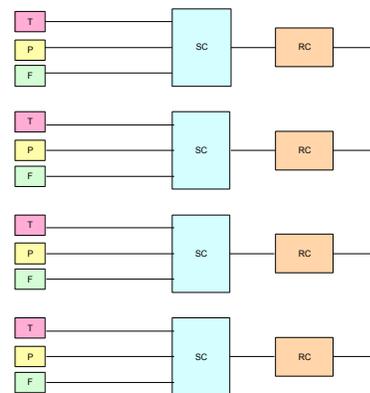
Hybrid redundancy techniques combine elements of both static and dynamic redundancy. In hybrid approaches, masking is used to prevent the propagation of errors, and error detection, diagnosis, and reconfiguration are used to handle faulty components.

## V. I&C ARCHITECTURE OPTIONS

The I&C system for the SNPP must successfully integrate all of the necessary sensors and actuators, their signal processing electronics, and the transmission of data between transducers and the control computers. Primary sensors included for this SNPP I&C architecture

evaluation are temperature (T), flux (F), and pressure (P) sensors. The process computers analyze the sensor data and if actions are required, transmit a signal to the actuators for the reactor control rod and/or reflector positioners, thermal transfer fluid pumps, and diversion or metering valves.

The simplest reactor controller (RC) design (Fig. 1) uses identical redundant sensors, RCs, hardware, and software. Redundant channel sensors relay a signal to a redundant signal conditioner (SC). Any of the three channel sensor outputs (P, T, or F) is sufficient for reactor control. Thus, all three sensors or the channel SC must fail for reactor control from an RC to be unavailable. Independent sensor inputs to each RC guarantee different input values to each RC because of sensitivity and calibration differences between the sensors and SCs. As sensor faults develop, isolating the failed sensor is essentially impossible because no comparative calculations between redundant channels are performed.



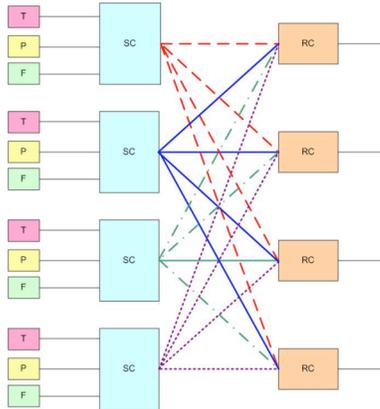
**Fig. 1. Independent sensor inputs and control outputs. (T, P and F are sensors, SC is signal controller, and RC is reactor controller)**

Two types of enhancements over the simplex system shown in Fig. 1 were evaluated:

1. modular redundant controllers, sensors, actuators, and communications; and
2. diverse controllers, sensors, actuators, and communications.

Sharing independent sensor values to all four RC channels should produce identical control calculations (Fig. 2). An identical algorithm calculates a consensus value in all four RC channels. Unlike the simplex design in Fig. 1, when a sensor fails, a common deviation checking algorithm should isolate the sensor fault. Even if all three sensors to a particular SC fail, the

corresponding RC should still be available to calculate consensus input values from shared sensor inputs from redundant channels. Penalties for being able to share sensor data include added mass and wiring complexity for point-to-point sensor value sharing. Software complexity also increases because of the ability to calculate consensus input value and to check input deviation.

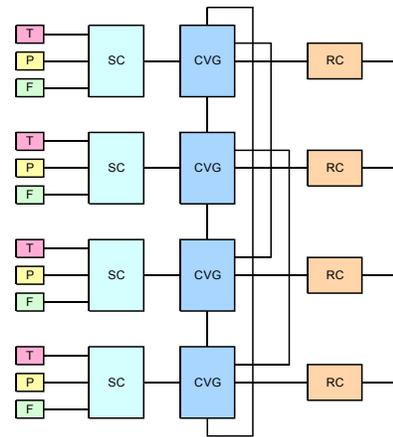


**Fig. 2. Independent sensor inputs shared between independent controllers.**

Adding cross-linking between common value generators (CVGs) allows shared sensor input values and consensus value cross checking between the independent RCs and sensor inputs (Fig. 3). This increases the likelihood that identical values are input to all redundant RC control calculations, thereby producing identical control calculation results. As sensors fail, common deviation checking algorithms in the CVGs isolate sensor faults. The loss of a common value generator or reactor controller is independent of the loss of sensors.

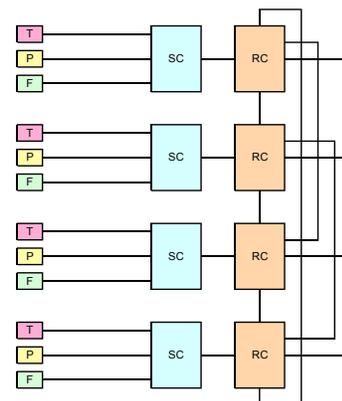
Inputting independent sensor value sets into redundant RC channels, and sharing the independent sensor values via cross-links to the RCs allows identical RC algorithms to calculate consensus sensor values in all RC channels (Fig 4).

The consensus sensor values should be identical and can be cross checked via RC cross-links. Because identical control algorithms calculate control outputs based on cross checked consensus sensor values, the RCs should produce identical control calculation results. Independent control calculation results can be cross checked via RC cross links, and any deviation of control calculation result indicates an RC fault. The loss of single sensor does not force the loss of an RC channel, however, the loss of single cross link forces multiple losses in RC channels.

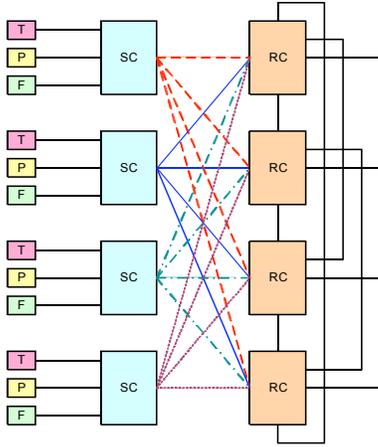


**Fig. 3. Independent sensor inputs and control outputs with cross-linked common value generators.**

Another similar architecture allows independent sensor values to be shared to all redundant RC channels (Fig. 5). Identical algorithms in each RC calculate consensus input values in all redundant RC channels. The consensus input values are cross-checked via RC cross-links. Control calculation in each RC should produce identical output results. Cross checking of control calculation results isolates RC faults. The loss of a sensor input (or link) does not result in the loss of an RC. However, the loss of cross-links could result in the loss of multiple RCs. In addition, single- or dual-failed sensors do not force RC faults but do allow ambiguous consensus input values as remaining sensors fail. Common deviation checking algorithms should isolate sensor faults.

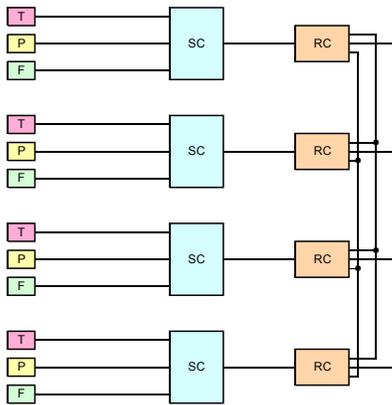


**Fig. 4. Independent sensor inputs to cross-linked controllers.**



**Fig. 5. Independent sensor inputs with shared values to cross-linked reactor controllers.**

In the last architecture reviewed, RC cross-links allow sensor input value sharing across channels to derive consensus input values and isolate sensor or SC faults (Fig. 6). Calculated consensus input values can be cross checked via the RC cross-links to ensure consistent inputs to independent control calculations. Control calculation results can be cross checked for consistency and RC faults via the cross-links. Dual cross-link buses allow single fault tolerance in RC cross-links.



**Fig. 6. Independent sensor inputs with dual-bus cross-linked reactor controllers.**

For each option that uses cross-linking, the systems are fault-tolerant in that failed RCs (e.g., CPUs) can be removed from the system. If output value comparators use voting logic, a 3-out-of-4 vote then becomes a 2-out-of-3 vote. If the failure of another RC results in only two operable RCs, the output values are compared. If the values differ significantly, the system removes one RC

from operation and operates on a single RC. This allows all sensor values to be available for comparative analyses, given that the sensor(s) are still operating.

## VI. QUANTIFICATION

When evaluating each of the I&C architectures, two types of failures were considered:

1. Type I – the system fails to generate a control signal when a control change is necessary, and
2. Type II – the system generates a false control signal when a control change is unnecessary.

### VI.A. Data

Quantification of reliability implies the existence of validated data for the hardware and software failures. Several sources of data for CPUs, buses, software, and sensor failures were collected from reports that analyzed spacecraft and aerospace systems [Ref. 4–10]. These values were then compared to failure rates for similar components in the chemical and nuclear industries [Ref. 11–18].

According to NASA’s Langley Research Center, “digital systems (both hardware and software) are notorious for their unpredictable and unreliable behavior” [4]. Even after the most thorough and rigorous testing, some bugs remain.

The results are not any better for terrestrial systems. A review of licensee event reports for operating nuclear power plants shows that software errors are the most likely cause of digital I&C system failures [19].

**TABLE I. Hardware and software failure rates**

Reference	Failure probability
CPUs	$2 \times 10^{-5}/h$
Busses	$1 \times 10^{-6}/h$
Software	$1 \times 10^{-3}$
Sensors	$5 \times 10^{-5}/h$

### VI.B. Common-Cause Failures (CCFs)

The defense-in-depth concept that has been applied to the design of nuclear power plant safety systems has created a situation in which the risk is determined by accidents that cannot occur unless multiple components fail to perform their design functions. Therefore, the risks posed by nuclear power plant operation are usually dominated by dependent failure scenarios. This statement

is supported by the results of probabilistic risk assessments (PRAs) and by the historical experience of nuclear power plant operation. Similarly, the control system for the SNPP will require multiple, independent failures to occur for the system to fail. Thus, system failure will also be dominated by dependent failure scenarios.

The alpha-factor model was used to estimate the common-cause failure probabilities that would normally be determined from a set of failure ratios and the total component failure rate. Because it is not always possible to determine parameters by analyzing operating data for all the components of interest, a set of “generic” alpha factor values was used [20].

## VII. RESULTS

In evaluating each of the alternative I&C architectures, the Type I failure, where the system fails to generate a trip signal given that a trip is necessary, was quantified. At the end of 5 years, the unreliability of the control system will be about 0.737 (Fig. 7), regardless of what design alternative is chosen.

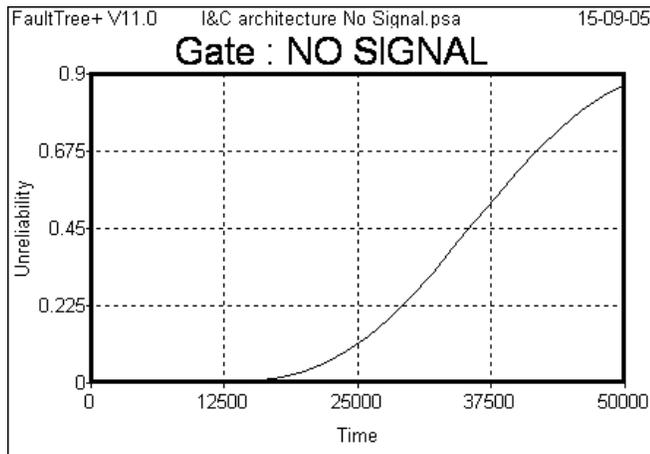


Fig. 7. Unreliability vs mission time.

All of the design options have about the same unreliability for several reasons. For long-term operation, N-MR/FT configurations were only equal to and potentially less reliable than simplex configurations in terms of wear-out. At first, this seemed strange in light of all the N-MR/FT control architectures used in high-reliability military, space, and general aviation systems. But the difference has to do with mission duration vs equipment wear-out duration. If mission duration is shorter than the 50% reliability lifetime of submodules, then adding modular redundancy increases reliability over

simplex operation during a mission. However, if mission duration is longer than the 50% reliability lifetime of individual modules, reliability decreases more rapidly for N-MR fault-tolerant (N-MR/FT) configurations than the functional equivalent simplex configuration.

Early on in the mission for each architectural option, causes of system failure will be dominated by common-cause failures (CCFs). For reliable systems with redundancies, this is always true. As the length of the mission increases (i.e., the time variable becomes larger), independent failures begin to dominate. At about 10,000 h, the contribution from the CCF of all four CPUs is about the same as that for the likelihood of all four CPUs failing independently. Thus, the importance of the CCFs and independent failures of the CPUs is about the same at ~10,000 h (Fig. 8). Around 15,000 h the sensors begin to appear in all of the dominant cut sets and thus become the most important components. At ~43,000 h, the independent failure of all 12 sensors accounts for over 30% of the failure probability.

Surprisingly, the independent failure of 12 components (the sensors) is 15 times more likely than the independent failure of 4 components (the CPUs). To explain this, consider a sensor with a failure rate of  $5 \times 10^{-5}/h$ . After an elapsed time of 43,800 h, the probability of failure for that component (i.e., the unreliability) is 0.8881, from

$$\bar{r} = 1 - e^{-\lambda t}, \quad (1)$$

where  $\lambda$  is the failure rate, and  $t$  is the elapsed mission time.

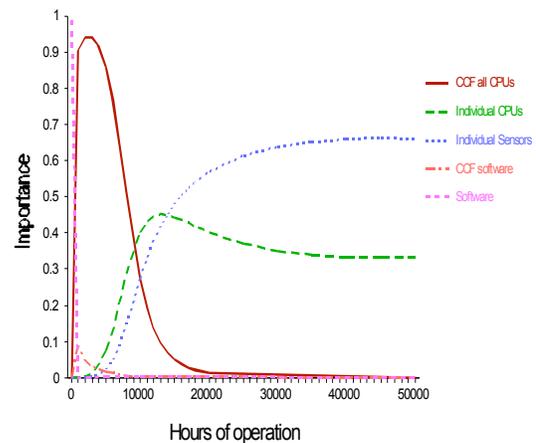


Fig. 8. Importance of basic events.

The independent failure probability of three sensors (T, P, and F) at 43,800 h is 0.7. All four sets of sensors must fail for the system to fail, or  $p = 0.24$ . Similarly, the probability of failure for a CPU with a failure rate of  $1 \times 10^{-5}/h$  at 43,000 h is 0.3547. The independent failure probability of a CPU at 43,800 h is 0.0158. Because the probability of failure monotonically increases over time, when the mission length becomes very large, the probability of failure becomes very large.

## VIII. RECOMMENDATIONS

For a passive redundancy system, the modules are replicated multiple times depending on the desired fault tolerance capability. The modules provide input to a selection mechanism (voter) to mask errors that reach the outputs of the modules. With a single voter, the voter becomes a single-point failure.

Moving the voters to the input of the modules eliminates the single-point failure in the single-voter system. This configuration protects the computations performed by the replicated components but requires that redundant components reading the outputs use the same approach to prevent the propagation of errors and single point of failure.

With an active redundancy approach using duplication with comparison, error detection is achieved by comparing the outputs of two modules performing the same function. If the outputs of the modules disagree, an error condition is raised followed by diagnosis and repair actions to return the system to operation. In a similar approach, only one module would actually perform the intended function with the other component being a dissimilar monitor that checks the outputs looking for errors.

These modules can be arranged in a self-checking pair configuration (or dual-dual configuration). In this configuration, the comparators perform the error detection function. Normally the output is taken from one of the pairs known as the primary pair, with the other pair acting as a spare or backup. When an error on the primary is detected, the spare is brought online, and the primary is taken offline for diagnosis and maintenance if necessary.

A hybrid redundancy using an N-modular masking configuration with spares combines the masking approach used in passive redundancy with the error detection, diagnosis, and reconfiguration used in dynamic approaches. In the hybrid configuration, when an error is detected, the faulty module is taken offline for diagnosis,

and a spare module is brought online to participate in the error-masking configuration. Although this configuration has better dependability characteristics than purely passive or active configurations, the cost and complexity are higher.

Input consistency checking design must guard against the "Byzantine General" problem of propagating common wrong inputs to identical control algorithm calculators. Fault-tolerant systems, although internally redundant, must deal with single-source information from the external world. For example, a flight control system is built around the notion of feedback from physical sensors, such as temperature or pressure sensors. Although these can be and usually are replicated, the replicates do not produce identical results. To use bit-by-bit majority voting, all of the computational replicates must operate on identical input data. Thus, the sensor values (the complete redundant suite) must be distributed to each processor in a manner that guarantees that all working processors receive exactly the same value, even in the presence of some faulty processors. This is the classic Byzantine General problem; algorithms to solve the problem are called Byzantine agreement algorithms [5].

For short missions requiring extremely high reliability, system failure (if it occurs) is likely to be caused by incomplete coverage rather than the depletion of spares. N-modular redundancy is better for such applications. Standby sparing is better for long missions where failure is likely to be caused by depleted spares.

Standby sparing provides a more reliable system for long missions because of the reduced failure rates of warm and cold spares compared to hot spares. That is, cold and warm spares have zero and reduced the failure rates, while hot spare experience the full component failure rate [21].

N-MR/FT is most appropriate for short-duration, time-critical flight operations that might be susceptible to momentary electrical upset or premature component failure. Standby redundancy is appropriate to extend operation beyond normal wear-out reliability times and where standby module switchover can be leisurely accomplished without catastrophic system failure during switch-over.

This is supported in several areas of literature and is supported in NASA history of computing in space. This is why deep space missions use standby redundancy and Flight Control Systems use N-MR/FT. Depending on the mission operations and control dynamics of the JIMO

SNPP, there may be the need for a hybrid, reconfigurable, modular redundant configuration with cold standby spare modules to tolerate momentary upset and to endure through the total mission lifetime.

Most of the reliability literature is concentrated on electrical components. Unfortunately, the highest stress for the JIMO SNPP is for most of the sensor and some of the actuator transducers. Because these components must be in physical contact with the extreme thermal and radiation operating environment of a minimally shielded reactor and the environment more or less causes the wear-out stress, these components must be considered to be in a static redundant configuration. To have a dynamic redundant configuration for sensors and actuators would require a mechanical means of changing the physical position of transducers. To a somewhat lesser degree, even the electronic I&C components inside the JIMO spacecraft electronics vault have a continuous radiation exposure stress throughout the mission and cannot receive the full benefit of “cold” sparing redundancy.

#### ACKNOWLEDGMENTS

This work was performed under the sponsorship of NASA’s Project Prometheus and directed by DOE/NNSA Naval Reactors. Opinions and conclusions drawn by the authors are not endorsed by DOE/NNSA Naval Reactors.

#### REFERENCES

- [1] S. V. Amari, H. Pham, and G. Dill, “Optimal Design of k-out-of-n:G Subsystems Subjected to Imperfect Fault-Coverage,” *IEEE Transactions on Reliability*, 53(4) (December 2004).
- [2] W. Torres-Pomales, *Software Fault Tolerance: A Tutorial*, NASA/TM-2000-210616, October 2000.
- [3] C. A. Liceaga and D. P. Siewiorek, *Automatic Specification of Reliability Models for Fault-Tolerant Computers*, NASA Technical Paper 3301, July 1993.
- [4] R. W. Butler et al., NASA Langley’s Research and Technology-Transfer Program in Formal Methods, May 2002.
- [5] M. V. Frank, “The Hessi Probabilistic Risk Assessment: A Risk Analysis of an Explorer Program Spacecraft,” Space Flight Safety, Proceedings of the Joint ESA-NASA Conference, June 11-14, 2002, Estec, Noordwijk, the Netherlands.
- [6] W. E. Vesely et al., *Fault Tree Handbook with Aerospace Applications*, Version 1.1, NASA, Washington, DC, August 2002.
- [7] P. Babcock, A. Schor, and G. Rosch, *Reliability Modeling Methodology for Independent Approaches on Parallel Runways Safety Analysis*, NASA/CR-1998-207660, April 1998.
- [8] R. Hemm and S. Houser, *A Synthetic Vision Preliminary Integrated Safety Analysis*, NS009S1, Logistics Management Institute, December 2000.
- [9] R. W. Butler and S. C. Johnson, *Techniques for Modeling the Reliability of Fault-Tolerant Systems With the Markov State-Space Approach*, NASA Reference Publication 1348, September 1995.
- [10] R. J. Bartos, “System Safety Analysis of an Autonomous Mobile Robot,” 12<sup>th</sup> International System Safety Conference, New Orleans, LA, July 5–10, 1994.
- [11] J. Hecht, A. T. Tai, and K. S. Tso, *Class 1E Digital Systems Studies*, NUREG/CR-6113, U.S. Nuclear Regulatory Commission, October 1993.
- [12] Greene and Bourne, *Reliability Technology*, Wiley & Sons, New York, 1972.
- [13] Southwest Research Institute, *Nuclear Plant Reliability Data System 1980 Annual Reports of Cumulative System and Component Reliability*, NUREG/CR-2232, September 1981.
- [14] *IEEE Guide to the Collection and Presentation of Electrical, Electronic, Sensing Component, and Mechanical Equipment Data for Nuclear-Power Generating Stations*, IEEE Std-500, Institute of Electrical and Electronic Engineers, 1984.
- [15] J. R. Welker and H. P. Schoor, “LNG Plant Experience Data Base,” AGA Transmission Conference, New Orleans, May 21-23, 1979.
- [16] *Development of an Improved LNG Plant Failure Rate Data Base* (March 1980–June 1981), GRI-80/0093, Gas Research Institute.
- [17] V. Skala, “Improving Instrument Service Factors,” *Instrumentation Technology*, November 1974.
- [18] E. J. Henley, H. Kumamoto, *Reliability Engineering and Risk Assessment*, Prentice-Hall, Englewood Cliffs, New Jersey, 1981.
- [19] R. T. Wood, “I&C Technologies for Advanced Reactors,” NRPCT Project Meeting, July 21, 2005.
- [20] A. Mosleh and D. M. Rasmuson, *Common Cause Failure Data Collection and Analysis System Volume 5—Guidelines on Modeling Common Cause Failures in Probabilistic Risk Assessments*, Draft, INEL-94-0064, Idaho National Engineering Laboratory, December 1995.
- [21] K. K. Vemuri, J. B. Dugan, and K. J. Sullivan, “Automatic Synthesis of Fault Trees for Computer-Based Systems,” *IEEE Transactions on Reliability*, 48(4) (December 1999).