ELSEVIER

Performance Evaluation 35 (1999) 215–232

PERFORMANCE EVALUATION
An International Journal

On-board preventive maintenance: a design-oriented analytic study for long-life applications

Ann T. Tai [a,*], Leon Alkalai [b], Savio N. Chau [b]

[a] *IA Tech, Inc. 10501, Kinnard Avenue, Los Angeles, CA 90024, USA*

[b] *Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA*

Abstract

With respect to the long-life missions associated with NASA's X2000 Advanced Deep-Space System Development Program, reliability implies a system's continuous operation for many years in an unsurveyed radiation-intense environment. Further, the stringent constraints on the mass of a spacecraft and the power on-board create unprecedented challenges on the means for achieving the ultra-high mission reliability. In this paper, we present an approach to on-board preventive maintenance which rejuvenates a system by letting system components rotate between on-duty and off-duty shifts, slowing down a system's aging process and thus enhancing mission reliability. By exploiting nondedicated system redundancy, hardware and software rejuvenation are realized simultaneously without significant performance penalty. Our design-oriented analysis confirms a potential for significant gains in mission reliability from on-board preventive maintenance and provides to us useful insights about the collective effect of age-dependent failure behavior, residual mission life, risk of unsuccessful maintenance and maintenance frequency on mission reliability.

Keywords: On-board preventive maintenance; Hardware and software rejuvenation; Phased-mission analysis; Mission reliability gain

1. Introduction

With NASA's spectacular return to Mars on 4 July 1997, the Mars Pathfinder Lander and its Sojourner Microrover have set a new standard for *Faster, Better, Cheaper* space exploration missions. The X2000 Advanced Deep-Space System Development Program will raise the standard even higher by providing to multiple long-life deep-space missions an engineering model equipped by a suite of new-generation space technologies [1]. Specifically, X2000 is aimed at achieving at least an order of magnitude improvement in both performance and dependability under stringent power and mass constraints, while enabling a high-level efficiency such that the cost of a multi-mission purpose spacecraft could be lower than that of the Mars Pathfinder spacecraft [2]. Currently, five missions are designated to be the recipients of

* Corresponding author. Tel.: +1-310-474-3568, fax: +1-310-474-3608; e-mail: a.t.tai@ieee.org

0166-5316/99/$ – see front matter
PII: S0166-5316(99)00006-1

the X2000 technologies: Europa Orbiter, Pluto–Kuiper Express, Solar Probe, Mars Sample Return and DS4/Champollion (also known as Comet Sample Return).

With respect to the X2000 long-life missions, reliability implies a system's continuous operation for many years in an unsurveyed deep-space radiation-intense environment. On the other hand, the stringent constraints on the mass of a spacecraft, the power on-board and the launch cost preclude traditional fault tolerance approaches which rely on extensive component/subsystem replication. In other words, the means for achieving the ultra-high reliability must emphasize the utilization of nondedicated system resource redundancy. Among other things, we have been investigating into the notion of *on-board preventive maintenance* [3]. By "on-board preventive maintenance", we mean the actions taken place during a mission for eliminating or minimizing potential error conditions that accrue over the operational life of a spaceborne system. Although the concept is similar to that of "software rejuvenation" which has received a significant amount of attention in the past few years [4–7], our investigation into on-board preventive maintenance takes one step further as we concern not only software but also hardware. Software and hardware rejuvenation can be realized simultaneously on board because:

- From software perspective, aging phenomenon such as memory leakage and data corruption can be removed via program reinitialization and/or system rebooting which clean up a system's internal state [4,5], resulting in *complete age reversal*.
- From hardware perspective, during a power-off period, the effects of electronmigration that occurs in microelectronics when current density is high can be reduced through structural/thermal relaxation [8,9], and the radiation damages that accrue through mission events such as gravitation assist can be mitigated by electron tunneling into the trapped positive charge [10,11]. These annealing mechanisms (see Section 2 for a more detailed explanation) will result in *partial age reversal*.

Accordingly, the procedure of on-board preventive maintenance for the X2000 computing system involves (1) stopping the running software and host hardware, and (2) rebooting the system and restarting software execution after a scheduled time interval. To minimize maintenance-caused system unavailability, we exploit nondedicated system redundancy. An instance of nondedicated redundancy in the X2000 system architecture is the following: During a critical mission phase which demands a full computation power (such as the Encountering Phase in the 15-year long Pluto–Kuiper Express mission), all the processor strings are scheduled to jointly perform spacecraft and scientific functions, while only a subset of the strings is mandated to be in service during less-critical mission phases such as a cruise phase. Hence, individual processor strings can rotate between on-duty and off-duty shifts based on a scheduled time interval, we call it a *duty period*, for preventive maintenance throughout the mission except during the phase(s) requiring a full computation power. In this manner, on-board preventive maintenance practically has no negative effect on system availability.

Our initial study demonstrated the feasibility of on-board preventive maintenance [3]. For simplicity, the analysis was based on the assumption that the aging processes of hardware and software components both could be completely reversed through preventive maintenance and thus could be treated the same in analytic evaluation. Although this assumption sufficed the purpose of our preliminary study, we have been investigating into this subject in further depth by discriminating between hardware and software with respect to the effects of preventive maintenance on them. In particular, we use Weibull distribution to characterize a system component's aging and age-reversal processes in a cohesive manner. We then derive a recursive function for mission reliability evaluation which captures the dependencies of system components' aging/failure behavior between duty periods. Further, we extend our basic model to facilitate phased-mission analyses. Inspired by the results of our earlier study which revealed that an optimal duty period is operational environment dependent [3], we utilize the extended model and the mission profiles of

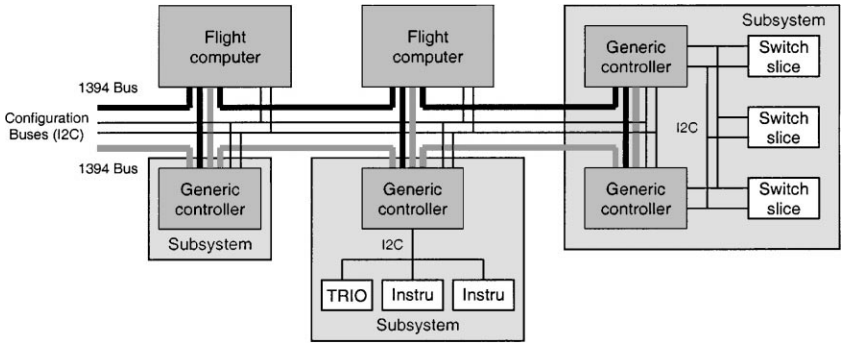

Fig. 1. X2000 system architecture.

Pluto–Kuiper Express and DS4/Champollion to investigate into the influence of phase-adjusted preventive maintenance on mission reliability gain. The evaluation results confirm a potential for significant gains in mission reliability from on-board preventive maintenance and provide to us useful insights about the collective effect of age-dependent failure behavior, residual mission life, risk of unsuccessful maintenance and maintenance frequency on mission reliability.

The rest of the paper is organized as follows. Section 2 provides the background information about the X2000 system architecture. Section 3 describes the method of model construction, followed by Section 4 which discusses the results of the analytic evaluation based on two X2000 mission profiles. Section 5 summarizes what we have accomplished and presents an outline for our future research.

2. Preventive maintenance in X2000 architecture

One of the major challenges the X2000 program exhibits to us is the requirements diversity among the five missions, which demand a computation power from a single processor string to multiple strings, a throughput range from under 20 MIPs to over 100 MIPs, and a mass memory size from 100 MB to 1.5 GB. Therefore, the X2000's computing system architecture must be scalable and distributed in order to accommodate a broad spectrum of requirements. As far as the avionics concern, the X2000 is aimed at further advancing the packaging technologies initiated by the New Millennium Deep Space One (NMP DS1) program [12–14]. The NMP DS1 developed an architecture which consists of a RAD-6000 processor multi-chip-module (MCM), a local memory MCM, a nonvolatile mass memory MCM, and an I/O MCM. The X2000 architecture has taken a further step toward miniaturization, in which each processor string consists of a processor slice integrated with I/O interfaces, a local memory slice, and one to four nonvolatile mass memory. Moreover, the X2000 architecture has been enhanced through employing standard commercial bus interfaces, namely, IEEE 1394 and I2C, with novel topologies for better performance and reliability and efficient power utilization [15,16]. As the use of standard bus interfaces enables the X2000 architecture to be adaptable to various mission requirements, the system can accommodate from a single to multiple processor strings. An instance of a two-string configuration of the X2000 architecture is depicted in Fig. 1.

A main feature of the X2000 system architecture is the I/O cross-strapping of the processor strings with the standard bus interfaces IEEE 1394 and I2C. The cross-strapping is implemented using redundant I/O ASICs as shown in Fig. 2. This feature permits the workload that comprises spacecraft and science functions to be shared by and migrated between processor strings in an efficient manner. Therefore, the I/O

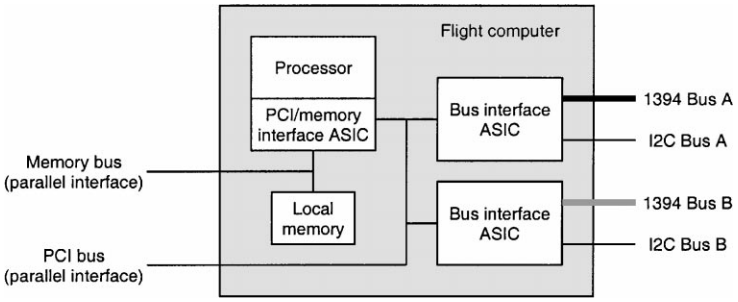

Fig. 2. I/O cross-strapping.

cross-strapping plays an important role in enabling adaptive utilization of system resource redundancies, a cost-effective way to the realization of dependability and performance enhancement. Among other things, this feature can be utilized to facilitate on-board preventive maintenance. That is, during less critical mission phases such as cruise phases, the processor strings can be scheduled on and off duty periodically, servicing the mission on a rotation basis, which enables (1) significant saving of the limited power on-board, and (2) periodic rejuvenation for both hardware and software of the processor strings.

As mentioned in Section 1, from hardware perspective, there are at least two mechanisms that allow material to be rejuvenated when power is not applied. The first mechanism is the annealing of electronmigration. This mechanism is particularly beneficial to space microelectronics. Specifically, the metal lines in microelectronics of a new-generation miniaturized spacecraft have extremely small cross-sections. Thus, these circuits carry very high density current, a condition usually conducive to electronmigration that causes voids (also called “vacancies”) in conductors. On the other hand, the annealing process that occurs during power-off periods of a long-life mission enables the electronmigration caused voids to be self-repaired through a structural/thermal relaxation process [8,9], acting as preventive maintenance to improve system lifetime. The second mechanism is the annealing of structural damages in semiconductors caused by radiation. This mechanism is especially important for deep-space missions that need *gravitation assist* from large planets such as Jupiter and Saturn. Those planets have very strong radiation regions that are formed by the interactions between their strong magnetic fields and the high-energy particles in space. During gravitation assist, the spacecraft has to pass through such environments and thus the semiconductors on-board could experience severe radiation damages. However, these damages can be annealed through *electron tunneling* into the trapped positive charge [10,11], which can occur during a power-off period (after gravitation assist) and function as preventive maintenance.

3. Methods of model construction

3.1. Problem description

The analytic models we develop in this section are based on the system configuration of the X2000 architecture with two processor strings (see Section 2, Fig. 1). We first construct a basic model that captures the essence of on-board preventive maintenance; we then extend the basic model for phased-mission analyses. Based on the extended model, we investigate into the preventive maintenance strategies using the mission profile of Pluto–Kuiper Express which is to explore the unsurveyed planet Pluto and

Kuiper Disk. As Pluto is the most distant planet in the Solar system, this mission will travel the greatest distance from Earth and have the longest duration among the X2000 missions (15 years). We carry out further investigation based on the profile of DS4/Champollion which is a sampling-return mission.

Due to the combined consideration of the time-increasing failure rate and partial age reversal, we face a challenge of representing the dependencies between duty periods. In particular, when a new duty period begins, the age of the string which just completes its maintenance and becomes power-on again is a function of (1) the accumulated amount of time comprising the duty periods the string has serviced since the mission starts, and (2) the amount of age reversal it has obtained from the prior preventive maintenance. Another difficulty in model construction arises from the fact that the continuation of the duty-switching sequence will depend upon the availability of resource redundancy. Specifically, resource redundancy may become temporarily or permanently unavailable for preventive maintenance if both strings are required to jointly service the mission or if one of the strings fails, respectively. Moreover, the simultaneous consideration of hardware and software rejuvenation requires us to (1) differentiate between hardware and software with respect to the effects of preventive maintenance on them (partial age reversal versus complete age reversal), and (2) capture the interactions between hardware and software in terms of their failure behavior. Finally, analyzing phase-adjusted on-board preventive maintenance needs us to deal with a "phase hierarchy" in the sense that each *duty period* can be viewed as a mission phase at the lower level while each actual *mission phase* at the higher level consists of duty periods. Although various approaches to phased-mission analysis were proposed by other researchers (see [17–19], for example), analytic models considering the requirements described above have not yet been investigated. In the sections that follow, we develop a model construction method in which:

1. Weibull distribution is utilized to characterize a system component's aging and age-reversal behavior in a cohesive manner.
2. A recursive function is derived to facilitate the representation of the dependencies between duty periods, with respect to the aging, age-reversal and failure behavior of system components.

Before we proceed to describe the model construction method in further detail, we explain our assumptions as follows:

1. In accordance with the theory that power-off periods will improve the lifetime of microelectronics [8], we postulate that the amount of age reversal obtained by the hardware of a string through preventive maintenance is directly proportional to the length of a power-off period [1] (which equals to the length of a duty period due to the rotation-based preventive maintenance scheduling).
2. Both hardware and software failures considered in the model are permanent in nature and will cause the corresponding string to be in a nonoperational state. When one of the strings fails, the surviving string will attempt to take over the workload. However, if both strings fail, the computing system will become nonoperational, which leads to an unsuccessful mission.
3. A string may crash when it attempts to takeover the workload from its peer during scheduled duty switching or during failure recovery, causing the system to be in a nonoperational state. We call this event "an unsuccessful duty switching" and use the term "switching coverage" to refer to the complement of the probability of such an event.
4. A string will not fail during its power-off periods.

[1] While the general trend is evident that power-off periods will repair electronmigration and radiation caused damages and lengthen the lifetime of microelectronics, researchers observed differing mathematical relationships between the amount of increased lifetime and duration of a power-off period [8]. For simplicity, we choose to use the linear relationship in this study but our analytic models can also accommodate other types of assumption.

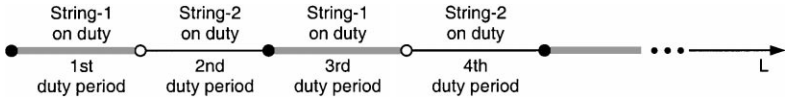

Fig. 3. Duty-switching sequence.

3.2. Characterizing aging and age-reversal processes using Weibull distribution

The Weibull distribution is the most commonly used distribution in reliability engineering because by a proper choice of its shape parameter, an increasing, decreasing or constant failure rate distribution can be obtained [20]. Weibull distribution has been used to describe system behavior with time-increasing failure rate such as fatigue failure and vacuum-tube failure [21]. In fact, Weibull distribution not only enables us to characterize the age-dependent failure rate of a system component by properly setting the shape parameter but also allows us to model the age-reversal effect from on-board preventive maintenance through the use of the "location parameter". Specifically, we can begin model construction with choosing the following form of Weibull probability density function [20]:

$$f(t) = \beta\lambda((t-\gamma)\lambda)^{\beta-1}\mathrm{e}^{-((t-\gamma)\lambda)^{\beta}}, \tag{1}$$

where β is the shape parameter (we set it to a value greater than 1 to represent the age-increasing failure rate), λ is the scale parameter and γ is the location parameter that defines the "origin" where the system begins to age and to have a potential for failure. To aid a more precise description, we depict a duty-switching sequence in Fig. 3 and define the following notation:

- S *service age* of a system component
- δ amount of age reversal resulting from a preventive maintenance

By "service age," we mean the accrued amount of time during which a processor string is on duty to perform spacecraft and science functions. As shown in Fig. 3 where the time horizon corresponds to calendar time L, the shaded regions will contribute to the service age of String-1. In these terms, the mathematical concepts for the aging and partial age-reversal processes of *a single string* can be illustrated in Fig. 4. In the figure, the abscissa marks the service age of the string while the ordinate measures $h_i(S)$, its Weibull failure rate (hazard rate) function for the ith duty period, namely,

$$h_i(S) = \beta\lambda((S-\gamma_i)\lambda)^{\beta-1}, \tag{2}$$

where γ_i equals to the total quantity of partial age reversal experienced by the string through the preventive maintenance prior to the ith duty period. Consequently, the term $(S-\gamma_i)$ in Eq. (2) represents the "effective age" of the string in the ith duty period. For clarity of illustration, the shape parameter β is set to 2 in this example such that $h_i(S)$ is linearly increasing within the ith duty period. Each solid dot marks the beginning of a duty period for the string (presuming that a duty period has a duration of 10 weeks in this example). The lines with arrows at their right-hand sides illustrate the effect of partial age reversal. More precisely, these lines indicate the following: By the time when the string is ready to start a new duty period, its age has been reversed, due to the effect of the preventive maintenance just completed, by δ time units (assuming five weeks in this example) as if its "birthday" (which is represented by the value of the location parameter γ_i in Eq. (2)) moves forward along the service-age horizon (such that the string becomes "younger"). Each thick solid line segment represents the effective failure rate of the string for a particular duty period. Finally, each dashed line with arrows at both ends measures the effective age of the string at the time when it starts a new duty period.

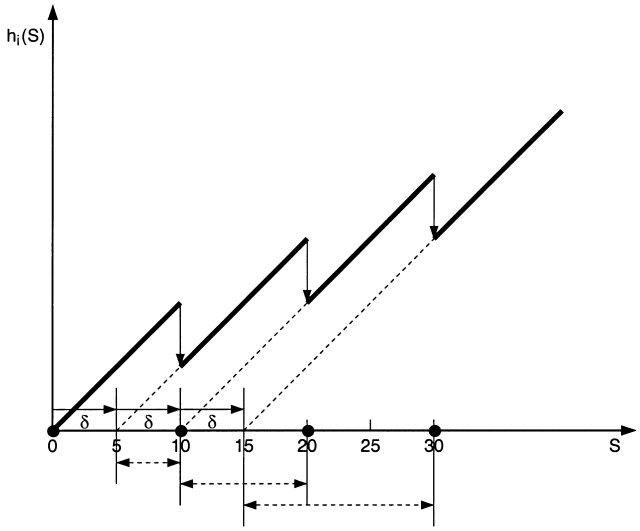

Fig. 4. Partial age-reversal concept from service-age perspective.

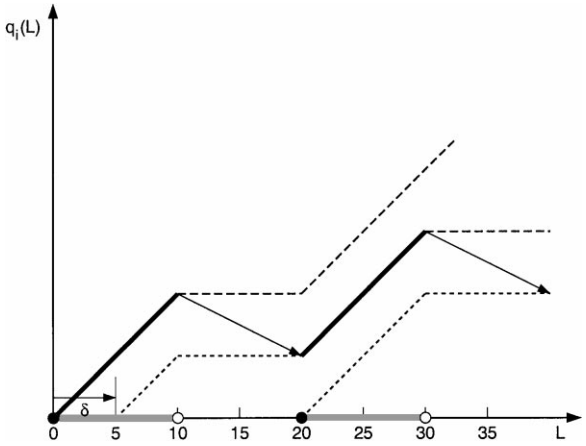

Fig. 5. Partial age-reversal concept from mission calendar-time perspective.

Fig. 5 shows the effective Weibull failure rate of a string from the perspective of mission calendar time L. In this figure, the solid dots mark the starting points of the duty periods of String-1, coinciding with the solid dots in Fig. 3. Similarly, the hollow dots mark those for String-2, coinciding with the hollow dots in Fig. 3. Note that the Weibull failure rate functions $h_i(S)$ and $q_i(L)$ (in which service age is expressed as a function of calendar time L) in Figs. 4 and 5, respectively, are equivalent for a particular duty period i. In fact we can view the composed solid curve representing the Weibull failure rate in Fig. 5 as the "stretched" version of that in Fig. 4.

3.3. Capturing dependencies between duty periods using a recursive function

Recursion methods are powerful means for representing dependencies between successive system events; accordingly, researchers developed a number of efficient recursion techniques to evaluate the

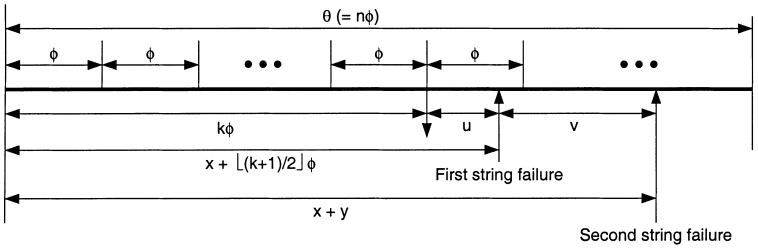

Fig. 6. Duty-period oriented timing diagram.

behavior of gracefully degradable fault-tolerant systems and effect of checkpointing mechanisms (see [22,23] for example). With regard to our application, we devise a recursive function based model to capture the dependencies of system behavior between duty periods, which is described below.

3.3.1. Basic model

The system's behavior with respect to duty periods is a regenerative renewal process [24], which can be translated into a duty-period oriented timing diagram shown in Fig. 6. The notation used in the figure are defined below:

- θ duration of a mission
- n number of duty periods in a mission
- ϕ duration of a duty period
- k number of duty periods with successful completions prior to the first string failure
- x hardware's service age of a string when its error condition causes the first string failure
- y hardware's service age of a string when its error condition causes the second string failure
- u software's service age of a string when its error condition causes the first string failure
- v software's service age of a string when its error condition causes the second string failure

The timing diagram illustrates the success and failure scenarios in terms of duty period and service ages of hardware and software, and describes the relationships between them. Namely,

1. $k\phi$ marks the first k duty periods through which both strings do not fail.
2. $x + \lfloor (k+1)/2 \rfloor \phi$ (or interchangeably, $k\phi + u$) is the time to the first string failure caused by a hardware or software error, where x and $\lfloor (k+1)/2 \rfloor \phi$ are the service ages of the failed and surviving strings at the time of the failure, respectively.
3. $x + y$ (or interchangeably, $k\phi + u + v$) is the time to the second string failure caused by a hardware or software error.

Based on the timing diagram, we can analyze the system's success and failure scenarios:

S1 $k = n \Rightarrow$ both strings do not fail during the mission

S2 $(k < n) \wedge (x + y > n\phi) \Rightarrow$ the first string failure occurs during the $(k+1)$th duty period due to a hardware or software error and the surviving string remains operational through the remainder of the mission

S3 $(k < n) \wedge (x + y \leq n\phi) \Rightarrow$ the first string failure occurs during the $(k+1)$th duty period due to a hardware or software error and the surviving string subsequently fails before the end of the mission

Let $R(\theta)$ denote the reliability of a mission with duration θ, clearly

$$R(\theta) = P[\mathrm{S1}] + P[\mathrm{S2}]. \tag{3}$$

Before we proceed to derive the solution for $R(\theta)$, we introduce the following notation:

- $T_1[i]$ service age of a string's hardware at the time when the string starts its duty for the ith duty period
- $T_2[i]$ service age of a string's hardware at the time when the string completes its duty for the ith duty period

From the diagram that illustrates the duty-switching sequence (Fig. 3), it follows that

$$T_1[i] = \left\lfloor \frac{i-1}{2} \right\rfloor \phi, \quad T_2[i] = \left(\left\lfloor \frac{i-1}{2} \right\rfloor + 1 \right) \phi.$$

We can then define $F[i]$ as the probability that a string fails during the ith duty period due to a hardware error, namely,

$$F[i] = \int_{T_1[i]}^{T_2[i]} f_i(x)\,\mathrm{d}x, \tag{4}$$

where f_i is the Weibull probability density function characterizing hardware's failure behavior in the ith duty period, that is,

$$f_i(t) = \beta\lambda((t-\gamma_i)\lambda)^{\beta-1}\mathrm{e}^{-((t-\gamma_i)\lambda)^{\beta}}, \tag{5}$$

where $\gamma_i = \lfloor (i-1)/2 \rfloor \delta$. And in accordance with the assumption that the amount of age reversal obtained by the hardware of a string through a preventive maintenance is directly proportional to the length of a duty period, we let $\delta = \rho\phi$, where the coefficient ρ has a domain $[0, 1)$.

Further, let $G[i]$ denote the conditional probability that a string fails due to a hardware error during the ith duty period given that both strings do not fail by the end of the $(i-1)$th duty period. $G[i]$ can be solved in terms of a recursive function that facilitates the representation of the dependencies between duty periods, with respect to strings' aging, age-reversal and failure behavior. More precisely,

$$G[i] = \frac{F[i]}{\prod_{j=1}^{i-1}(1-G[j])}, \quad i \geq 2 \tag{6}$$

with

$$G[1] = F[1].$$

Although there are alternative ways to formulate $R(\theta)$, we choose to employ the recursive function because it is easy to understand and facilitates model extension (as described in Section 3.3.2). Based on $G[i]$, the first term in Eq. (3) can be evaluated by a product-form expression,

$$P[\mathrm{S1}] = c^{n-1} \prod_{i=1}^{n} (1-G[i])(1-Q[i]), \tag{7}$$

where c is the "switching coverage" (see Section 3.1) and $Q[i]$ is the probability that a string fails due to a software error during the ith duty period. Since software is able to obtain a complete age reversal through rejuvenation, the formulation for $Q[i]$ is simpler

$$Q[i] = \int_0^{\phi} \hat{f}(t)\,\mathrm{d}t, \tag{8}$$

where $\hat{f}$ is the Weibull probability density function with the shape parameter α and scale parameter μ that characterizes software's failure behavior

$$\hat{f}(t) = \int_0^{\phi} \alpha\mu(t\mu)^{\alpha-1}\mathrm{e}^{(t\mu)^{\alpha}}\,\mathrm{d}t.$$

The derivation for the solution of the second term in Eq. (3) requires us to consider the interactions between hardware and software with respect to their failure behavior [25]. To aid the formulation, we introduce the following notation:

- A_i event that both strings do not fail by the end of the ith duty period
- B_i event that a string failure caused by a hardware error occurs during the ith duty period and the surviving string remains operational through the remainder of the mission
- C_i event that a string failure caused by a software error occurs during the ith duty period and the surviving string remains operational through the remainder of the mission

Finally, we let

$$H[k] = P[A_k \cap B_{k+1}] = P[A_k]P[B_{k+1}|A_k],$$

$$S[k] = P[A_k \cap C_{k+1}] = P[A_k]P[C_{k+1}|A_k].$$

According to its definition, $P[A_k]$ can be expressed as

$$P[A_k] = c^k \prod_{i=1}^{k}(1 - G[i])(1 - Q[i]).$$

Based on the information supplied by the duty-period oriented timing diagram (Fig. 6), the solutions for $P[B_{k+1}|A_k]$ and $P[C_{k+1}|A_k]$ can be obtained by analyzing the strings' service ages and residual mission life at the time of failure. More precisely,

$$P[B_{k+1}|A_k] = \frac{1}{\prod_{i=1}^{k}(1 - G[i])}\int_{T_1[k+1]}^{T_2[k+1]} f_{k+1}(x)V_1(k,x)V_2(k,x)V_3(k,x)\,\mathrm{d}x, \tag{9}$$

where

$$V_1(k,x) = 1 - \frac{1}{\prod_{i=1}^{k}(1 - G[i])}\int_{\lfloor (k+1)/2\rfloor\phi}^{\theta-x} f_{k+1}(y)\,\mathrm{d}y,$$

$$V_2(k,x) = 1 - \int_0^{x-T_1[k+1]} \hat{f}(t)\,\mathrm{d}t,$$

$$V_3(k,x) = 1 - \int_0^{\theta-(x+\lfloor (k+1)/2\rfloor\phi)} \hat{f}(t)\,\mathrm{d}t.$$

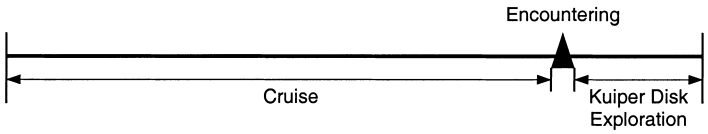

Fig. 7. Mission phases of Pluto–Kuiper Express.

Note that the above terms can be viewed as functions of k and x, the number of duty periods through which both strings do not fail and the service age of the string's hardware at the time of its failure during the $(k+1)$th duty period, respectively. Likewise, we have

$$P[C_{k+1}|A_k] = \int_0^{\phi} \hat{f}(u) W_1(k,u) W_2(k,u) W_3(k,u)\, \mathrm{d}u, \tag{10}$$

where

$$W_1(k,u) = 1 - \int_0^{\theta-(k\phi+u)} \hat{f}(v)\, \mathrm{d}v,$$

$$W_2(k,u) = 1 - \frac{1}{\prod_{i=1}^{k}(1-G[i])} \int_{T_1[k+1]}^{T_1[k+1]+u} f_{k+1}(x)\, \mathrm{d}x,$$

$$W_3(k,u) = 1 - \frac{1}{\prod_{i=1}^{k}(1-G[i])} \int_{\lfloor (k+1)/2 \rfloor \phi}^{\theta-(T_1[k+1]+u)} f_{k+1}(y)\, \mathrm{d}y.$$

Note that u is the service age of the software at the time of its failure during the $(k+1)$th duty period, whereas $T_1[k+1]+u$ corresponds to the service age of the host hardware. To this end, the second term of Eq. (3) can be given by

$$P[\mathrm{S2}] = \sum_{k=0}^{n-1} c(H[k]+S[k]) = \sum_{k=0}^{n-1} c P[A_k](P[B_{k+1}|A_k] + P[C_{k+1}|A_k]). \tag{11}$$

In turn, the measure we seek to evaluate, $R(\theta)$, can then be solved analytically.

3.3.2. Phased-mission analysis

Now we extend our basic model described in Section 3.3.1 for a phased-mission analysis. Consider the Pluto–Kuiper Express consisting of three phases, namely, the Cruise Phase, Encountering Phase and Kuiper-Disk Exploration Phase, as shown in Fig. 7, with phase durations of 12 years, 4 months and 3 years, respectively. Among the three phases, the Encountering Phase during which the spacecraft flies by Pluto is clearly the most critical to the mission. The crucial spacecraft functions include orbit maneuver and pointing, while those intensive scientific functions encompass taking high-resolution visible and infrared images, uplinking radio-science phase shift data and storing them on-board for later transmission. Therefore, as mentioned previously, both strings will be powered on to perform spacecraft and science functions in this phase. After encountering Pluto, the mission enters the Kuiper-Disk Exploration Phase during which the spacecraft will attempt to encounter one or more Kuiper Disk objects as it continues its journey out into interstellar space. This phase is aimed at enabling the long-life mission to gain potential bonus. Due to this nature, the Kuiper-Disk Exploration Phase does not mandate a full computation power. Accordingly, on-board preventive maintenance will be allowed to resume during this final phase.

The notation used in describing the phased-mission analysis is defined below:

- n_1 number of duty periods in the Cruise Phase
- n_2 number of duty periods in the Kuiper-Disk Exploration Phase
- ϕ_1 duration of a duty period in the Cruise Phase
- ϕ_2 duration of a duty period in the Kuiper-Disk Exploration Phase
- Φ duration of the Encountering Phase
- δ_1 amount of age reversal a string obtains from a preventive maintenance in the Cruise Phase
- δ_2 amount of age reversal a string obtains from a preventive maintenance in the Kuiper-Disk Exploration Phase

If further, we let θ_1 and θ_2 denote the durations of the Cruise Phase and Kuiper-Disk Exploration Phase, respectively, then

$$\phi_1 = \frac{\theta_1}{n_1}, \quad \phi_2 = \frac{\theta_2}{n_2} \quad \text{and} \quad \delta_1 = \rho\phi_1, \quad \delta_2 = \rho\phi_2,$$

where $\rho \in [0, 1)$.

The fact that the Encountering Phase requires both strings to be in service implies that preventive maintenance will temporarily halt. Therefore, special treatment is required for that particular phase. To preserve the generality of the equations developed for the basic model described in Section 3.3.1, we view the Encountering Phase as a special phase which can be "unfold" to become two "parallel" duty periods. More precisely, at the lower level of the model where strings' aging and failure behavior are represented, we treat the parallel duty periods as two individual duty periods; on the other hand, at the higher level where mission reliability is formulated, we view them as a single duty period. In this manner, the basic model for solving $R(\theta)$ can be adapted to accommodate phased-mission analyses with some minor modifications. First, we re-formulate γ_i, $T_1[i]$ and $T_2[i]$ by considering the parities of n_1 and i, and their relationships with the strings' duty-switching scheduling:

$$\gamma_i = \begin{cases} \left\lfloor \frac{i-1}{2} \right\rfloor \delta_1, & i \le n_1, \\ \left\lfloor \frac{i-2}{2} \right\rfloor \delta_1, & i \in \{n_1+1, n_1+2\}, \\ \left\lfloor \frac{n_1}{2} \right\rfloor \delta_1 + \left\lfloor \frac{(i-(n_1+2))-1}{2} \right\rfloor \delta_2, & n_1 \in \{2l \mid l = 1, 2, \dots\}, \\ & i \in \{n_1 + (2l+1) \mid l = 1, 2, \dots\}, \\ \left\lfloor \frac{n_1-1}{2} \right\rfloor \delta_1 + \left\lfloor \frac{i-(n_1+2)}{2} \right\rfloor \delta_2, & \text{otherwise}, \end{cases}$$

$$T_1[i] = \begin{cases} \left\lfloor \frac{i-1}{2} \right\rfloor \phi_1, & i \le n_1 + 2, \\ \left(\left\lfloor \frac{n_1}{2} \right\rfloor + 1 \right) \phi_1 + \Phi + \left\lfloor \frac{(i-(n_1+2))-1}{2} \right\rfloor \phi_2, & n_1 \in \{2l+1 \mid l = 0, 1, 2, \cdots\}, \\ & i \in \{n_1 + 2(l+2) \mid l = 0, 1, 2, \dots\}, \\ \left\lfloor \frac{n_1}{2} \right\rfloor \phi_1 + \Phi + \left\lfloor \frac{(i-(n_1+2))-1}{2} \right\rfloor \phi_2, & \text{otherwise}, \end{cases}$$

$$T_2[i] = \begin{cases} \left(\left\lfloor \frac{i-1}{2} \right\rfloor + 1 \right) \phi_1, & i \le n_1, \\ \left\lfloor \frac{i-1}{2} \right\rfloor \phi_1 + \Phi, & i \in \{n_1+1, n_1+2\}, \\ \left(\left\lfloor \frac{n_1}{2} \right\rfloor + 1 \right) \phi_1 + \Phi + \left(\left\lfloor \frac{(i-(n_1+2))-1}{2} \right\rfloor + 1 \right) \phi_2, & n_1 \in \{2l+1 \mid l = 0, 1, 2, \dots\}, \\ & i \in \{n_1 + 2(l+2) \mid l = 0, 1, 2, \dots\}, \\ \left\lfloor \frac{n_1}{2} \right\rfloor \phi_1 + \Phi + \left(\left\lfloor \frac{(i-(n_1+2))-1}{2} \right\rfloor + 1 \right) \phi_2, & \text{otherwise}. \end{cases}$$

Table 1
Parameter value assignment for Pluto–Kuiper Express study

θ_1	θ_2	Φ	β	λ	α	μ	ρ	c
624	156	17.3	5.5	0.0005	5.0	0.001	0.50	0.9999999

Based on the above modifications, Eq. (6), the recursive function, can be adapted to facilitate a phased-mission analysis by deriving just one more boundary condition for handling the Encountering Phase which comprises two "parallel" duty periods

$$G[i] = \frac{F[i]}{\prod_{j=1}^{i-1}(1 - G[j])}, \quad i \geq 2, \quad i \neq n_1 + 1, \tag{12}$$

with

$$G[n_1 + 1] = \sum_{i=n_1+1}^{n_1+2} \frac{F[i]}{\prod_{j=1}^{n_1}(1 - G[j])}$$

and

$$G[1] = F[1].$$

Letting $n = n_1 + n_2 + 1$, reliability of the Pluto–Kuiper Express mission can then be evaluated using Eqs. (3), (7) and (11).

4. Evaluation and discussion

4.1. Pluto–Kuiper Express study

Applying the model developed in Section 3 and using *Mathematica*™, the effectiveness of on-board preventive maintenance is evaluated based on the mission profile of Pluto–Kuiper Express. It is worth to note that the recursive function $G[i]$ and conditional expressions for $T_1[i]$ and $T_2[i]$ in the model can easily lend themselves to efficient computer manipulation by utilizing the built-in recursion capability and conditional constructs of *Mathematica*™. First, we study the influence of phase-adjusted maintenance frequency on reliability gain. That is, mission reliability $R(\theta)$ is evaluated along two dimensions – against varying maintenance frequencies for the Cruise Phase and Kuiper-Disk Exploration Phase (n_1 and n_2, respectively). The value assignment for other parameters is shown in Table 1, where all the parameters involving time (durations, rates, etc.) presume that time is quantified in weeks.

Table 2 displays mission reliability $R(\theta)$ as a function of n_1 and n_2, where $R_0(\theta)$ denotes the "baseline mission reliability" – assuming the system does not have preventive maintenance during the entire mission. And Fig. 8 provides a graphical presentation of the evaluation results, where n_1 and n_2 are plotted in a logarithmic scale. These illustrations confirm a potential for significant gains in mission reliability from on-board preventive maintenance, namely, an improvement up to two orders of magnitude relative to the baseline reliability can be accomplished. On the other hand, the evaluation reveals that extremely high maintenance frequencies will not be effective for mission reliability enhancement, as indicated by the entries of the row and column where $n_1 = 1000$ and $n_2 = 1000$, respectively. This can be understood by considering the tradeoffs between system component reliability improvement due to preventive

Table 2
Evaluation results of Pluto–Kuiper Express study, $R_0(\theta) = 0.9995874562$

n_1	n_2				
	1	10	50	100	1000
1	0.9995885863	0.9996769935	0.9996800733	0.9996763954	0.9995955107
5	0.9999888363	0.9999879463	0.9999839475	0.9999789486	0.9998889706
20	0.9999969549	0.9999960546	0.9999920547	0.9999870548	0.9998970611
100	0.999989273	0.9999883728	0.9999843729	0.9999793731	0.9998893799
1000	0.9998993555	0.9998984554	0.9998944559	0.9998894565	0.9997994713

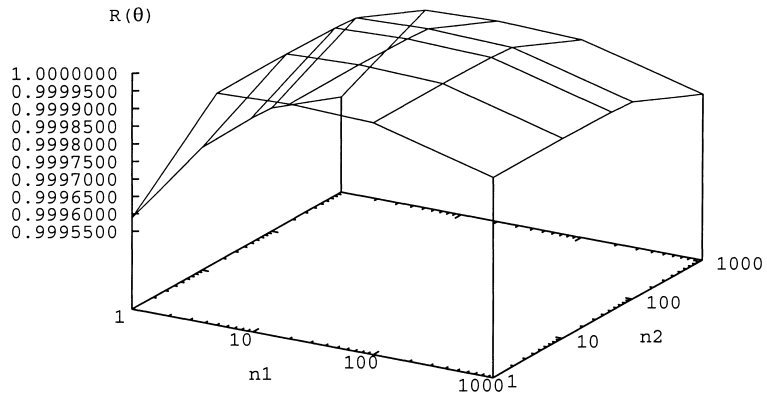

Fig. 8. $R(\theta)$ as a function of n_1 and n_2.

maintenance and the likelihood of system failure caused by unsuccessful duty switching. In particular, an extremely high maintenance frequency will lead to excessive risk of unsuccessful duty switching which negates the potential benefit from preventive maintenance.

A more interesting observation with regard to adjusting maintenance frequency to mission phase is the following: Intuitively, a later mission phase favors more frequent maintenance (thus shorter duty periods) due to a higher vulnerability of system failure derived from component aging. Contrarily, our evaluation results reveal that it will not be beneficial in general to increase maintenance frequency (i.e., to decrease the duration of a duty period) in the later mission life. Indeed, a strategy that exercises preventive maintenance in a less frequent manner in the later mission life will usually lead to an optimal mission reliability. As indicated in Table 2, for the particular mission in question, the optimal mission reliability will be achieved when $n_1 = 20$ and $n_2 = 1$, corresponding to the duty periods of 31.2 and 156 weeks, respectively. In other words, the evaluation results suggest to us that not to resume preventive maintenance after the Encountering Phase will indeed benefit mission reliability. This surprising result stems from some tradeoffs among system attributes which may not be obvious without analytic modeling. Specifically, the likelihood that the system fails before the mission completion tends to (1) increase as the age-dependent failure rates of system components increase, and (2) decrease as the residual mission life decreases. In other words, while the system becomes more vulnerable to failure as its components are aging, the decreasing residual mission life favors less frequent maintenance because the reliability improvement becomes less significant and may turn to be inadequate to compensate the risk of mission failure caused by unsuccessful duty switching.

Table 3
Evaluation with a higher switching coverage, $R_0(\theta) = 0.9995874839$

n_1	n_2				
	1	10	50	100	1000
1	0.9995886853	0.9996779012	0.9996845743	0.9996853881	0.9996853485
5	0.9999893313	0.9999893321	0.9999892926	0.9999892426	0.9999883429
20	0.9999989349	0.9999989256	0.9999988856	0.9999988356	0.9999979356
100	0.999999173	0.9999991637	0.9999991238	0.9999990738	0.9999981738
1000	0.9999983504	0.9999983412	0.9999983012	0.9999982512	0.9999973512

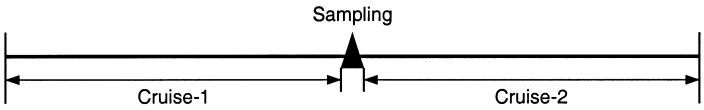

Fig. 9. Mission phases of DS4/Champollion.

Table 4
Parameter value assignment for DS4/Champollion study

θ_1	θ_2	Φ	β	λ	α	μ	ρ	c
182	182	2	5.5	0.0005	5.0	0.0025	0.50	0.9999999

Next we analyze the effect of switching coverage on optimal preventive maintenance frequency by using the set of parameter values shown in Table 1 but increasing the switching coverage c to 0.999999999. The evaluation results are displayed in Table 3. Contrast this table with Table 2, it can be observed that mission reliability improvement becomes more significant, namely, three orders versus two orders of magnitude with respect to the optimal mission reliability gains. Whereas the optimal preventive maintenance frequency for the Cruise Phase is now increased to 100, corresponding to a duty period of 6.2 weeks (versus 31.2 weeks). Nonetheless, the optimal preventive maintenance frequency for the Kuiper-Disk Exploration Phase remains 1, which again implies that it will not be beneficial to resume duty-switching after entering the final mission phase (although the reliability gain decreases in a less sensitive manner as n_2 increases, compared with the results shown in Table 2).

4.2. DS4/Champollion study

To further validate our findings, we apply the model to DS4/Champollion, another X2000 mission mentioned earlier. This mission is designed to visit and study Comet Temple 1 and return a sample to Earth. DS4/Champollion has a 7-year duration. Due to the sample-return nature of this mission, the most critical mission phase – Sampling Phase which demands a full computation power – will reside in the mid-point of the mission as shown in Fig. 9.

Tables 4 and 5 show the parameter value assignment for the DS4/Champollion study and the corresponding evaluation results, respectively. The results reveal that, preventive maintenance frequencies $n_1 = 20$ and $n_2 = 10$ (corresponding to the duty periods of 9.1 and 18.2 weeks, respectively) will lead to the optimal mission reliability. Thus, unlike Pluto–Kuiper Express, DS4/Champollion favors resuming preventive maintenance in the final mission phase (with a reduced frequency). Explanation for

Table 5
Evaluation results of DS4/Champollion study, $R_0(\theta) = 0.9984515096$

n_1	n_2				
	1	10	50	100	1000
1	0.9981746323	0.9990384087	0.9990346626	0.9990297612	0.9989415062
2	0.9997738205	0.9997742641	0.9997702753	0.9997652815	0.9996753961
20	0.9999966237	0.9999970704	0.9999930767	0.9999880767	0.9998980819
100	0.9999886464	0.999989093	0.9999850994	0.9999800995	0.9998901053
1000	0.9998986516	0.9998990982	0.9998951049	0.9998901055	0.9998001194

the distinction resides in the differing mission profiles. Specifically, in DS4/Champollion, the residual mission life after the Sampling Phase weighs 50% of the total mission duration, while in Pluto–Kuiper Express the residual mission life after the Encountering Phase weighs only about 20% of the total mission duration. The vulnerability of failure due to component aging in a phase whose duration weighs more significantly with respect to the mission life will have a greater negative impact on mission reliability. As a result, for DS4/Champollion, this vulnerability overweighs the risk of unsuccessful duty-switching and thus necessitates preventive maintenance in the final phase. In other words, whether it is beneficial to conduct preventive maintenance and how often to conduct it depend upon the tradeoffs between the two types of risks described above. Note that the conclusions of our studies are very consistent. That is, the evaluation results from all the studies show significant reliability gains from on-board preventive maintenance and suggest to us to exercise preventive maintenance with a reduced frequency or to stop preventive maintenance in the later mission life.

5. Conclusion and future work

We have obtained some useful results from the analytic studies of on-board preventive maintenance for long-life deep-space missions. Our model-based evaluation not only confirms the effectiveness of preventive maintenance but also provides to us further insights regarding the tradeoffs among system and environment attributes and their collective effect on mission reliability gain. From model construction perspective, we have proposed a novel use of Weibull distribution for (1) characterizing system components' aging and age-reversal processes in a cohesive manner, and (2) differentiating between hardware and software with respect to the effects of preventive maintenance on them. The recursive function developed in this paper can be utilized for phased-mission analyses for a variety of space applications, where system attributes have interdependencies between mission phases. For very-long life missions such as Pluto–Kuiper Express, further mission reliability gain may be accomplished through gradual maintenance frequency reduction. For example, the 12-year long Cruise Phase can be divided into a number of segments; maintenance frequency can be adjusted when the mission enters into a subsequent segment. To validate the hypothesis, we plan to conduct analytic and simulation studies. We are also motivated to investigate into the schemes that will further utilize the X2000 architecture's scalability such that the individual strings will be allowed to have various modes of rejuvenation, for example, to operate in different reduced power levels and to accommodate degradable computation quality. We plan to extend the analytic methods presented in this paper for analyzing those more sophisticated on-board preventive maintenance strategies.

While we are continuing our study for on-board preventive maintenance, we have been investigating into maintenance issues in a broader scope. Similar to the conventional notion of system maintenance, on-board maintenance collectively refers to preservation or improvement, during its operational life, of a system's ability to deliver a service complying with mission requirements. Related issues include evolvability, which permits a spaceborne system, during its mission's long life span, to keep pace with the latest technologies for better performance, fault tolerance and functionality, instead of being constrained by those available prior to mission launch. While evolvability itself can be viewed as on-board perfective maintenance, it necessitates corrective maintenance for detecting and tolerating potential inconsistencies between the old and new system configurations or software versions. Accordingly, how to adapt fault tolerance, dependability and performance engineering techniques to on-board maintenance is a subject of our future research.

Acknowledgements

The authors are thankful to the anonymous reviewers for their helpful comments. This research was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] L. Alkalai, NASA center for integrated space microsystems, in: Proceedings of Advanced Deep Space System Development Program Workshop on Advanced Spacecraft Technologies, Pasadena, CA, June 1997.
[2] L. Alkalai, A.T. Tai, Long-life deep-space applications, IEEE Comput. 31 (1998) 37–38.
[3] A.T. Tai, S.N. Chau, L. Alkalai, H. Hecht, On-board preventive maintenance: Analysis of effectiveness and optimal duty period, in: Proceedings of the Third International Workshop on Object-Oriented Real-time Dependable Systems (WORDS'97), Newport Beach, CA, February 1997, pp. 40–47.
[4] Y. Huang, C. Kintala, N. Kolettis, N.D. Fulton, Software rejuvenation: Analysis, module and applications, in: Digest of the 25th Annual International Symposium on Fault-Tolerant Computing, Pasadena, CA, June 1995, pp. 381–390.
[5] S. Garg, A. Puliafito, M. Telek, K.S. Trivedi, Analysis of software rejuvenation using Markov regenerative stochastic Petri net, in: Proceedings of the Sixth International Symposium on Software Reliability Engineering, Toulouse, France, October 1995, pp. 180–187.
[6] S. Garg, A. Puliafito, M. Telek, K.S. Trivedi, On the analysis of software rejuvenation policies, in: Proceedings of the 12th Annual Conference on Computer Assurance (COMPASS'97), Gaithersberg, MD, June 1997.
[7] S. Garg, A. Puliafito, M. Telek, K.S. Trivedi, Analysis of preventive maintenance in transaction based software systems, IEEE Trans. Comput. 47 (1998) 96–107.
[8] P.S. Ho, T. Kwok, Electromigration in metals, Rep. Progr. in Phys. 52 (1989) 301–348.
[9] K.-N. Tu, J.W. Mayer, L.C. Feldman, Electronic Thin Film Science for Electrical Engineers and Materials Scientists, Macmillan, New York, 1992.
[10] P.J. Rudeck, Long-term annealing of a radiation-hardened 1.0 micron bulk CMOS process, IEEE Trans. Nucl. Sci. 39 (1992) 1903–1911.
[11] T. Carriere, J. Beaucour, A. Gach, B. Johlander, L. Adams, Dose rate and annealing effects on total dose response of MOS and bipolar circuits, IEEE Trans. Nucl. Sci. 42 (1995) 1567–1574.
[12] L. Alkalai, J. Klein, M. Underwood, The New Millennium Program microelectronics systems, advanced technology development, in: Proceedings of the 34th Aerospace Science Meeting and Exhibit, Reno, Nevada, January 1996.
[13] L. Alkalai, A roadmap for space microelectronics technology into the New Millennium, in: Proceedings of the 35th Space Congress, Cocoa Beach, FL, April 1998.
[14] K. Sasidhar, L. Alkalai, A. Chatterjee, Testing NASA's 3D-stack MCM space flight computer, IEEE Design & Test Comput. 15 (1998) 44–55.

[15] S.N. Chau, X2000 avionics system conceptual design document, JPL Technical Report, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 1997.
[16] S.N. Chau et al., X2000 Architecture Tiger Team meeting review, Technical Report, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, June 1998.
[17] A. Bondavalli, I. Mura, M. Nell, Analytical modeling and evaluation of phased-mission systems for space applications, in: IEEE High-Assurance Systems Engineering Workshop, Washington, DC, August 1997.
[18] A.K. Somani, K.S. Trivedi, Phased-mission system analysis using Boolean algebraic methods, in: Proceedings of 1994 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, Nashville, TN, May 1994, pp. 98–107.
[19] J.F. Meyer, Performability evaluation of the SIFT computer, IEEE Trans. Comput. C-29 (1980) 501–509.
[20] D. Kececioglu, Reliability Engineering Handbook, vol. I, Prentice-Hall, Englewood Cliffs, NJ, 1991.
[21] K.S. Trivedi, Probability & Statistics with Reliability, Queueing, and Computer Science Applications, Prentice-Hall, Englewood Cliffs, NJ, 1982.
[22] E.G. Coffman, E.N. Gilbert, Optimal strategies for scheduling checkpoints and preventive maintenance, IEEE Trans. Reliability R-39 (1990) 9–18.
[23] J.L. Bruno, E.G. Coffman, J.C. Lagarias, P.W. Shor, Processor shadowing: Maximizing expected throughput in fault-tolerant systems, Math. Oper. Res., to appear.
[24] S.M. Ross, Stochastic Processes, 2nd ed., Wiley, New York, 1996.
[25] A.T. Tai, J.F. Meyer, A. Avižienis, Software Performability: From Concepts to Applications, Kluwer Academic Publishers, Boston, MA, 1996.

Ann T. Tai received her Ph.D. in computer science from the University of California, Los Angeles. She is the President and a Sr. Scientist of IA Tech, Inc., Los Angeles, CA. Prior to 1997, she was associated with SoHaR Incorporated as a Sr. Research Engineer. She was an Assistant Professor at the University of Texas at Dallas during 1993. Her research interests include development and application of performance, dependability and performability models for distributed systems, and fault-tolerant system architecture design. She authored the book, Software Performability: From Concepts to Applications, published by Kluwer Academic Publishers.

Leon Alkalai is the Center Director for the Center for Integrated Space Microsystems, a Center of Excellence at the Jet Propulsion Laboratory, California Institute of Technology. The main focus of the center is the development of advanced microelectronics, micro-avionics, and advanced computing technologies for future deep-space highly miniaturized, autonomous, and intelligent robotic missions. He joined JPL in 1989 after receiving his Ph.D. in computer science from the University of California, Los Angeles. Since then, he has worked on numerous technology development tasks including advanced microelectronics miniaturization, advanced microelectronics packaging, reliable and fault-tolerant architectures. He was also one of the NASA appointed co-leads on the New Millennium Program Integrated Product Development Teams for Microelectronics Systems, a consortium of government, industry, and academia to validate technologies for future NASA missions in the 21st century.

Savio N. Chau is a system engineer at the Jet Propulsion Laboratory. He is currently developing scalable multi-mission avionics system architectures for the X2000 Program. He has been investigating techniques to apply low-cost commercial bus standards and off-the-shelf products in highly reliable systems such as long-life spacecraft. His research areas include scalable distributed system architecture, fault tolerance, and design-for-testability. He received his Ph.D. in computer science from the University of California, Los Angeles. He is a member of Tau Beta Pi and Eta Kappa Nu.