On-Board Preventive Maintenance: Analysis of Effectiveness and Optimal Duty Period

Ann T. Tai † Savio N. Chau ‡ Leon Alkalaj ‡ Herbert Hecht †

† SoHaR Incorporated
Beverly Hills, CA 90211

‡ Jet Propulsion Laboratory
Pasadena, CA 91109

Abstract

To maximize reliability of a spacecraft which performs long-life (over 10-year), deep-space mission (to an outer planet), a fault-tolerant environment incorporating on-board preventive maintenance is highly desirable. In this paper, we present an initial model-based study which identifies the key factor for the reliability gain from on-board preventive maintenance and demonstrates the capability of analytic modeling in determining the optimal interval between maintenance (duty period).

1 Introduction

Pluto Express is a NASA mission to explore Pluto, the only unsurveyed planet in our solar system. Currently, Jet Propulsion Laboratory is performing studies to achieve the objectives of the mission. Due to the immense distance of Pluto, Pluto Express has very long mission life (12 years) which has created many unprecedented challenges [1]. For example, in order to reduce the flight time, the mass of Pluto Express has to be very low. Consequently, Pluto Express will have very limited power on-board. Furthermore, the reliability of the spacecraft is extremely demanding due to the long mission life. In order to meet these challenging requirements, the Pluto Express Data System employs an adaptive fault-tolerant architecture, in which two processor strings are able to share workload in a non-redundant manner [2]. Upon failure of one of the processor strings, the surviving string will takeover all the workload. To further enhance mission reliability, the design team has been investigating the notion of *on-board preventive maintenance*, which can be realized in a cost-effective manner based on the inherent system redundancy (the dual processor strings that perform spacecraft and scientific functions during encounter time). With on-board preventive maintenance, the two processor strings are scheduled to be on/off duty periodically, in order to reduce the likelihood of system failure due to radiation damage and other reversible aging processes. Moreover, since both the system and application software will be reinitialized when a string is powered on, switching between strings also results in *software rejuvenation*. The notion of software rejuvenation has been recently proposed for avoiding failures caused by potential error conditions accrued in the system environment such as memory leakage, unreleased file locks and data corruption [3]. The implementation of this idea involves deliberately stopping the running program and cleaning its internal state through flushing buffers, garbage collection, reinitializing the internal kernel tables or, more throughly, rebooting a computer. Such preventive maintenance procedures may result in appreciable system downtime, however, by exploiting the inherent hardware redundancy, the performance cost for our application could be minimal because 1) normally one of the strings will be performing its duty and, 2) the performance overhead for a string's re-initialization can be masked by starting it before the current active string gets off duty.

An essential issue in preventive maintenance is to determine the optimal interval between successive maintenance activities to balance the risk of system failure due to component fatigue/aging against that due to unsuccessful maintenance itself. Accordingly, we have been conducting a model-based dependability analysis, aimed at predicting the effectiveness of the on-board preventive maintenance approach and identifying the optimal *duty period* (we will use this term to refer to the interval between successive switching in the remainder of the paper). Due to the deterministic nature of a duty period, the system behavior cannot be directly represented by a Markovian process. However, via a hierarchical decomposition, we are able to construct and solve the analytic models in a relatively straight forward and simple manner. The results provide to us some useful insights. First, a sufficiently high switching coverage, defined as the prob-

0-8186-8046-6/97 $10.00

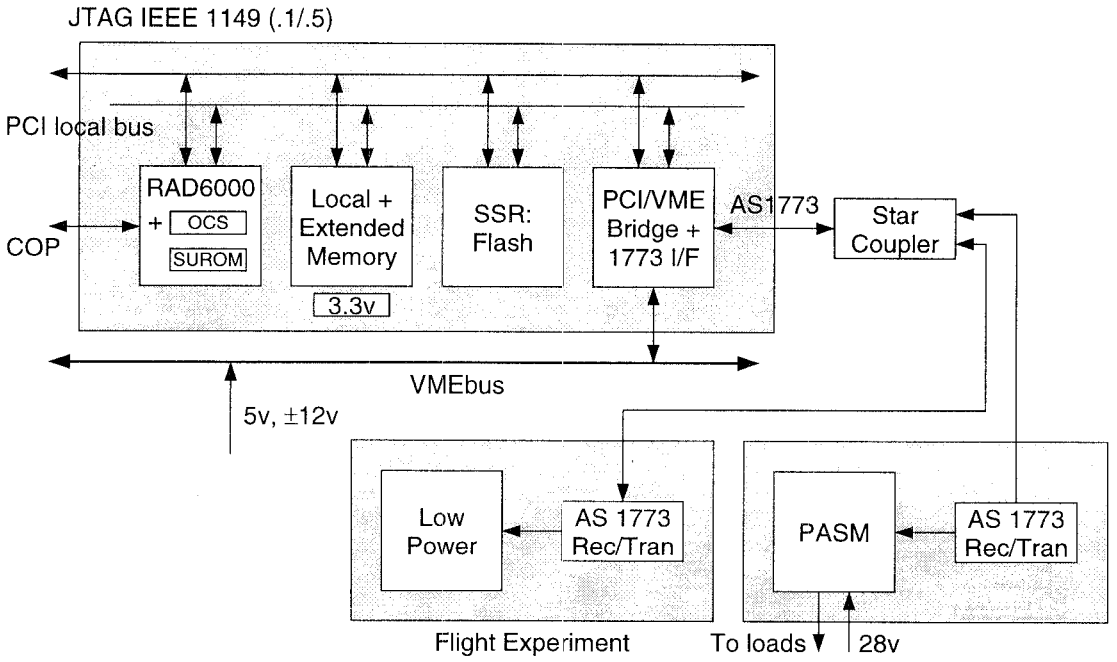

Figure 1: DS1 Architecture

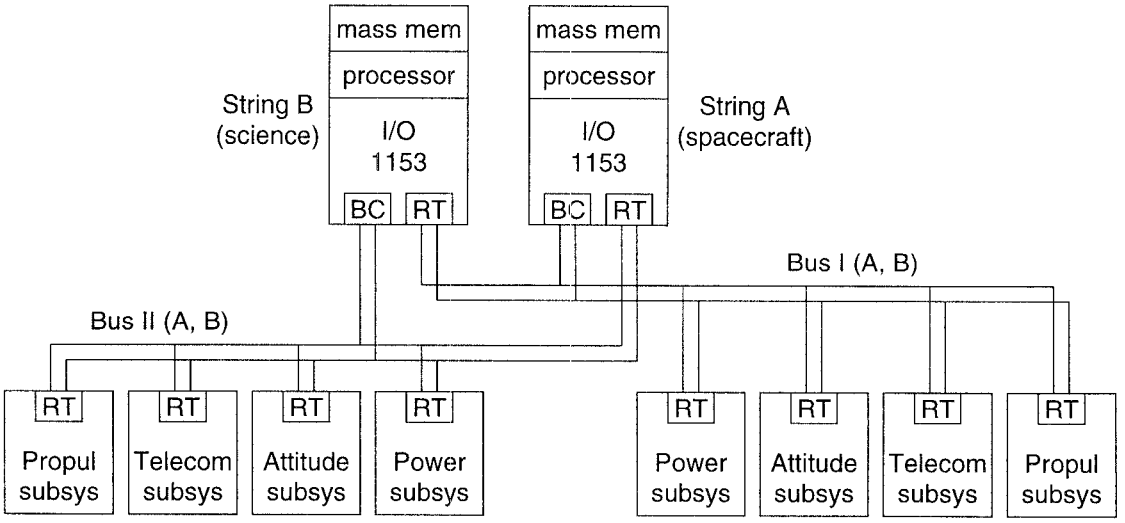

Figure 2: Pluto Express Data System Architecture

ability that the process of duty-shift from one string to the other is failure-free, is crucial for assuring mission-reliability gain from on-board preventive maintenance. Second, analytic modeling is indeed feasible for predicting optimal duty period.

Since we emphasize a methodological point of view rather than definitive numerical results based on accurate parameter values at this initial-study stage, the central purpose of this paper is to show how analytic modeling can be employed in guiding the design of on-board preventive maintenance. The remainder of the paper is organized as follows. Section 2 provides more background information about the Pluto Express Data System. Section 3 describes the method for model construction, followed by Section 4 which discusses the preliminary evaluation results. The concluding section summarizes what we have accomplished and discusses our plan for the future research.

2 Background

Pluto Express has adopted the technologies developed by the New Millennium Deep Space One (NMP DS1) program extensively [4, 5]. The NMP DS1 has developed an architecture which consists of a RAD-6000 processor multi-chip-module (MCM), a local memory MCM, a non-volatile mass memory MCM, and an I/O MCM (see Figure 1).

The MCMs are stacked together and are connected by the industrial standard PCI bus via vertical connectors. The Pluto Express Data System has extended this architecture by employing dual DS1 MCM stacks (referred to as processor strings hereafter) to enhance the system reliability. The main feature of the Pluto Express Data System's architecture is the I/O cross-strapping for the dual processor strings. This technique exploits the features of the 1553B protocol chips to achieve increased fault protection without adding much wiring complexity to the data interface (see Figure 2). Each processor string has its own 1553B bus and an additional interface to the 1553B bus of the other string. Further, the system design team propose to turn on only one processor string during the cruise phase, which will not only conserve power but also slow down the strings' aging process. Such a low-power operation is supported by the cross-strapping architecture described above. Clearly, the data interface architecture provides great flexibility to the preventive-maintenance oriented role switching between the strings.

We have conducted initial studies on the effectiveness of on-board preventive maintenance and optimal duty period based on two types of assumption regarding the failure behavior of a string. Namely,

- Staged failure process

  As assumed by [3], it takes time for a system or a process to "age" and then eventually crash. This implies the following: after a system starts its duty period, there is an interval during which the system is highly robust and very unlikely to fail; however, as potential error conditions such as memory leakage and data corruption accumulate, the system becomes vulnerable to failure. Assuming the times to vulnerable and failed stages are exponentially distributed, then the staged failure process is a Markov process as shown in Figure 3 and the time to failure is thus characterized by a hypoexponential distribution.

- Weibull distribution

  It is a common distribution used in reliability engineering [6] for modeling the effect of "aging" (time-increasing failure rate) or "maturing" (time-decreasing failure rate).

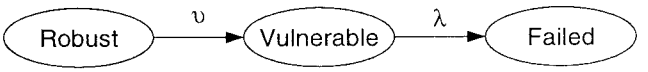

Figure 3: Staged Failure Process

3 Model Construction

A typical research issue on preventive maintenance is to find the optimal interval between successive maintenance activities which achieves high dependability without excessive cost associated with system down-time due to normal maintenance and system failure resulting from an unsuccessful maintenance. Earlier work related to identifying the optimal maintenance policy either used 1) a continuous-time Markov chain, assuming the interval between successive maintenance is exponentially distributed [3], or 2) a Markov regenerative stochastic Petri net (MRSPN), assuming a deterministic maintenance interval [7]. While the former is unrealistic, the later requires complicated and very time-consuming solution methods. Although Petri-net based modeling tools have been developed to accommodate deterministic transition time, to our best knowledge, they facilitate only steady-state solutions which are not meaningful to our application. On the other hand, the characteristics of the Pluto Express application allow us to employ a model construction method which is relatively simple. That is, we decompose the object system into two layers with respect to its temporal dimension — the lower layer represents the success/failure process in terms of duty period

while the upper layer represents the mission-level success/failure outcome. The approach is flexible in the sense that it allows us to vary basic assumptions about the failure process by modifying only the lower layer. The solution method is relatively simple because it involves only standard mathematical functions which can be implemented in general-purpose programming languages or using general-purpose mathematical software package such as Mathematica[1]. We describe the hierarchical model construction method below.

We start with representing the staged failure process of the dual-string system by a state-transition diagram as shown in Figure 4 (where we assume that the switching process takes a negligible amount of time). Each of the states in the diagram is encoded by two indicators which represent the status of the first and second strings, respectively. The definitions of the indicators are as follows.

- 1 A string is active and robust.
- 1′ A string is active and vulnerable to failure.
- 2 A string is in a "rejuvenation" mode.
- 0 A string has failed.

And the following are the definitions of the transitions:

- T_1 From a robust mode to a vulnerable mode.
- T_2 From a vulnerable mode to a failed mode but the other string takes over successfully.
- T_3 From a vulnerable mode to a failed mode and the system is unable to recover due to exhaustion of resources or unsuccessful switching.
- T_4 From a robust or vulnerable mode to a "rejuvenation" mode (switching).

Note that the corresponding process is not Markovian because the time to transition T4 is deterministic (scheduled switching), which suggests applying MRSPN for solution. However, a closer look at the characteristics of the problem leads us to consider an approach that enables us to obtain the desired measures in a more efficient manner. That is, as mentioned in the opening section, hardware redundancy (dual-string and I/O cross-strapping) allows the system downtime due to preventive maintenance be

[1]Mathematica is a registered trademark of Wolfram Research.

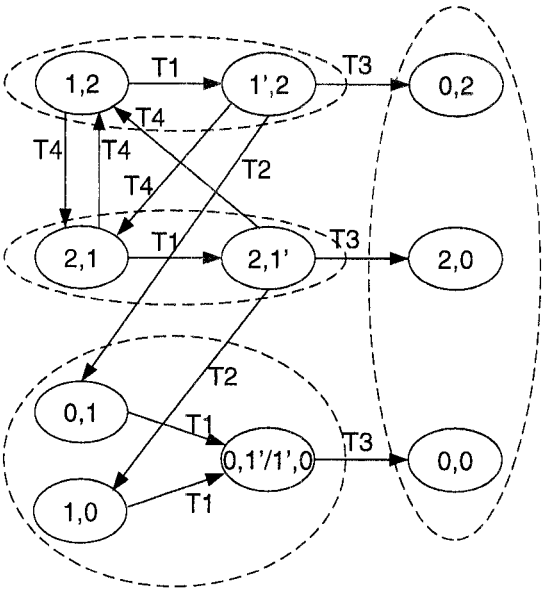

Figure 4: State-Transition Diagram

masked such that we are able to employ a non-state-space approach as described below. As the first step, we convert the state-transition diagram in Figure 4 into a cyclic series-parallel graph that represents system behavior from a duty-period perspective as shown in Figure 5, where each stage corresponds to a cluster of states (as indicated by those dashed ovals in Figure 4).

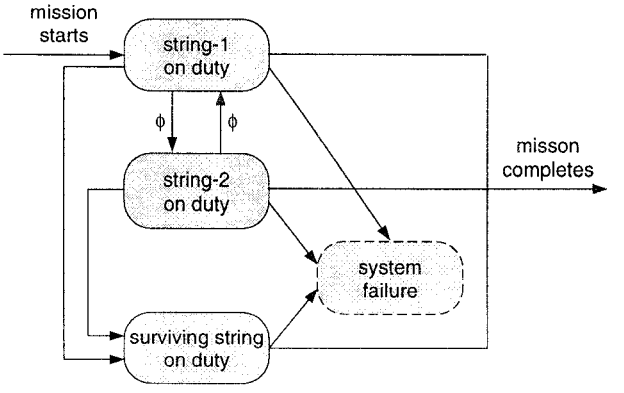

Figure 5: System Behavior

This series-parallel graph reveals that, at a higher level, the system's behavior with respect to the duty periods is a regenerative renewal process [8]. Accordingly, we can further translate the series-parallel graph into a duty-period oriented timing diagram describing the renewal process as depicted in Figure 6. The notation used in the timing diagram are defined below:

- ϕ The time duration of a string's duty period.
- n The switching frequency in per-mission dimension (thus $(n+1)$ is the number of duty periods in a mission).
- k The number of successful duty periods (a string does not fail during its duty period and the switching process at the end of the duty period is successful).
- x The time for the first string to reach a vulnerable mode.
- y The time for the first string to reach a failed mode from a vulnerable mode.
- z The time for the second string to reach a vulnerable mode.
- u The time for the second string to reach a failed mode from a vulnerable mode.

Thus we can analyze the system's success/failure scenarios in terms of the above notation as follows (see Figure 6).

- $k = n+1 \Rightarrow$ The mission succeeds with both strings being operational throughout the mission.
- $k < n+1 \wedge z+u > (n+1-k)\phi - x - y \Rightarrow$ One string fails during the $(k+1)^{th}$ duty period and the other string remains operational through the remainder of the mission.
- $k < n+1 \wedge z+u \leq (n+1-k)\phi - x - y \Rightarrow$ One string fails during the $(k+1)^{th}$ duty period and the other string subsequently fails before the end of the mission.

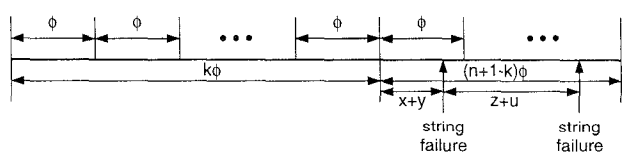

Figure 6: Duty-Period Oriented Timing Diagram

Letting θ denote the duration of a mission (θ = 12 year for the Pluto Express Data System), then $\phi = \theta/(n+1)$. If further, we let $R_\theta(\phi)$ denote the mission reliability with a duty period ϕ, it follows that

$$R_\theta(\phi) = c^n(1-F(\phi))^{n+1} + \sum_{k=0}^{n}(c(1-F(\phi)))^k c F_1(\phi, n, k) \quad (1)$$

where c is the coverage of switching process (the likelihood of a successful switching), F is the probability that a string fails before the end of a duty period via the stage of vulnerability, and F_1 is the conditional probability that a string fails (through a fatigue mode) in the $(k+1)^{th}$ duty period but the other string remains operational through the remainder of the mission given that the takeover switching process is successful. To solve for F and F_1, the probabilistic measures of the strings' behavior with respect to the time slots illustrated in Figure 6, we can utilize the Markov chain shown in Figure 7 (which is imbedded in the state-transition diagram in Figure 4).

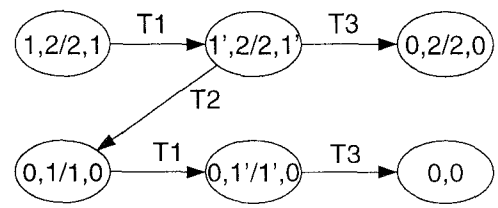

Figure 7: Lower-Layer Model

Although the measures can be obtained using standard transient solution method for continuous-time Markov chains, we choose to solve them through convolution which leads to a better understanding of the system behavior as one relates the lower-layer representation (Markov chain) to the duty-period oriented timing diagram. More precisely,

$$F(t) = \int_0^t g(x) \int_0^{t-x} h(y)\, dy\, dx \quad (2)$$

and,

$$F_1(t,n,k) = \int_0^t g(x) \int_0^{t-x} h(y) W(x,y,n,k)\, dy\, dx \quad (3)$$

where g and h are the density functions (pdfs) of the times for a string to reach its vulnerable and failed states, respectively (see Figure 3); whereas

$$W(x,y,n,k) = 1 - \int_0^{(n+1-k)t-x-y} g(z) \int_0^{(n+1-k)t-x-y-z} h(u)\, du\, dz$$

Therefore, the first term in Equation (1) corresponds to the probability that no failure occurs in any

of the $(n+1)$ duty periods and the switching processes are successful throughout the mission, and the second term summarizes the probabilities that a string failure occurs at the $(k+1)^{th}$ duty period but the other string successfully takes over and remains operational through the remainder of the mission. Note that, when $n = 0$, Equation (1) is reduced to

$$R_\theta(\theta) = (1-F(\theta))+cF_1(\theta,0,0) = 1-(F(\theta)-cF_1(\theta,0,0))$$

which exactly corresponds to the degenerate case in which on-board preventive maintenance is absent. Accordingly, we use $R_\theta(\theta)$ to denote the baseline mission reliability.

To this end, an optimal duty period that maximizes mission reliability can be defined by the following equation,

$$U(R_\theta(\phi) \mid D) = \max_{\phi\in(0,\theta]}\{\gamma\} \tag{4}$$

where D is a given set of system conditions (e.g., failure rate, switching coverage, etc.), and γ is a reward function through which mission reliability $R_\theta(\phi)$ is formulated (i.e., Equation (1)). Thus, optimizing mission reliability for a given set of system conditions D corresponds to maximizing γ with respect to the candidate duty periods with durations in the domain $(0,\theta]$.

If we assume that the time to failure has a Weibull distribution (instead of assuming a staged-failure process), the upper-layer model remains the same (thus Equation (1) is still valid) because the mission success criteria are independent of the low-level component-failure characterization. However, the lower-layer representation does change such that

$$F(t) = \int_0^t f(x)dx \tag{5}$$

and,

$$F_1(t) = \int_0^t f(x)V(x,n,k)\,dx \tag{6}$$

where f is the pdf of the time to failure, that is,

$$f(t) = \alpha\lambda(\lambda t)^{\alpha-1}e^{-(\lambda t)^\alpha}$$

and

$$V(x,n,k) = 1 - \int_0^{(n+1-k)t-x} f(y)\,dy$$

4 Discussion

Applying the models described in the previous section, the effectiveness of on-board preventive maintenance is evaluated with respect to mission-reliability gain from preventive maintenance and optimal duty period is also studied. As the first step, we study mission reliability under the assumption of staged-failure process. Figures 8, 9 and 10 depict mission reliability $R_\theta(\phi)$ as a function of switching frequency n for different system parameters ($\phi = \theta/(n+1)$, ν is the rate for a string going from its robust state to vulnerable state, λ is the failure rate[2] and $R_\theta(\theta)$ is the corresponding baseline mission reliability). We observe the following (for all three cases, the likelihood of an unsuccessful switching $(1-c)$ is set to 10^{-8}):

- When $\nu = 0.001$ and $\lambda = 0.0001$, preventive maintenance can increase mission reliability from 0.999952 to the 0.999997 range (about one order); and the optimal duty period is 6.2 weeks ($n = 100$).
- When the failure rate λ is doubled, preventive maintenance can increase mission reliability from 0.999812 to the 0.999995 range (about two orders); and the optimal duty period is 3.1 weeks ($n = 200$).
- When the failure rate λ is tripled, preventive maintenance can increase mission reliability from 0.999587 to the 0.999993 range (about two orders); and the optimal duty period is 2.1 weeks ($n = 300$).

The results demonstrate that in the postulated environment on-board preventive maintenance indeed increases mission reliability given that the switching coverage c is sufficiently high (equivalently speaking, the uncoverage $(1-c)$ is sufficiently low). Further, the curves reveal the influence of failure rate on optimal duty period. That is, the higher the failure rate, the shorter the duty period should be. This is a reasonable result because a less reliable system in general requires more frequent maintenance.

Figure 11 displays the results of an evaluation in which we assume that the failure process of a string is characterized by a Weibull distribution. The results are consistent with those from the analyses based on staged-failure assumption. From this curve, we see an improvement of mission-reliability about three orders (increased from 0.997213 to the 0.999997 range) and the optimal duty period is 4.2 weeks ($n = 150$).

We have also conducted analyses for the effect of the likelihood of an unsuccessful switching $(1-c)$ on mission reliability gain from preventive maintenance. Tables 1 and 2 display the numerical results (based on

[2]Both ν and λ have per-week dimensions.

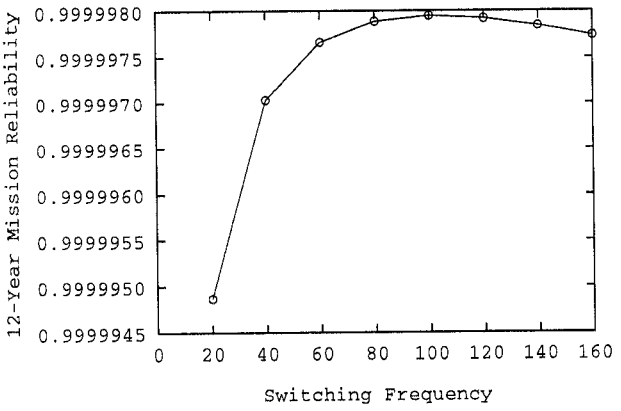

Figure 8: Optimal Switching Frequency
($\nu = 0.001$,$\lambda = 0.0001$,$R_\theta(\theta) = 0.999952$)

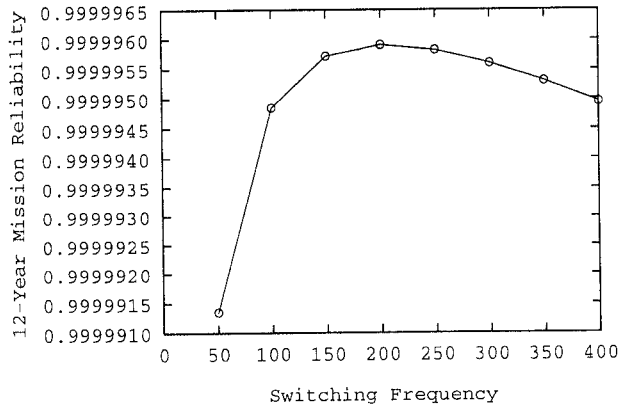

Figure 9: Optimal Switching Frequency
($\nu = 0.001$,$\lambda = 0.0002$,$R_\theta(\theta) = 0.999812$)

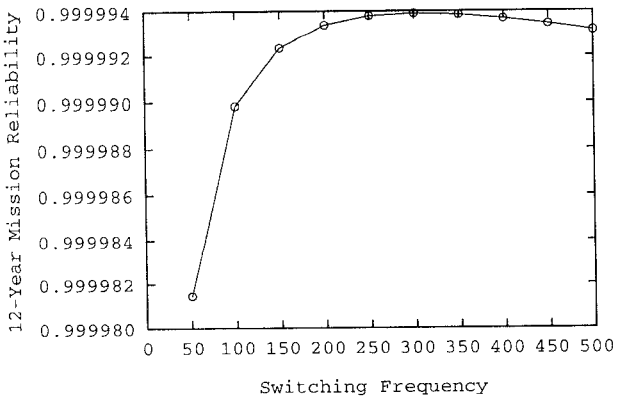

Figure 10: Optimal Switching Frequency
($\nu = 0.001$,$\lambda = 0.0003$,$R_\theta(\theta) = 0.999587$)

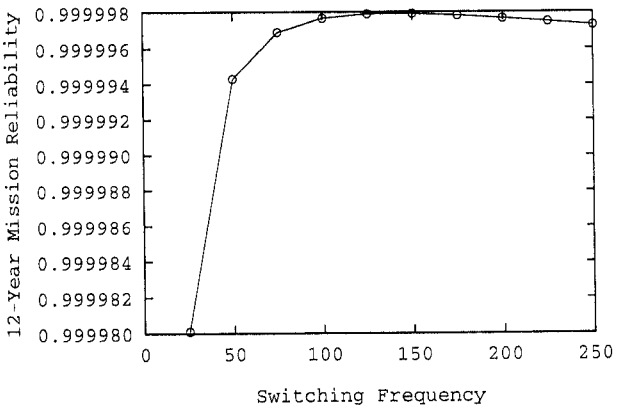

Figure 11: Optimal Switching Frequency
($\alpha = 3.0$,$\lambda = 0.001$,$R_\theta(\theta) = 0.997213$)

the staged-failure assumption). Except $(1-c)$ being a variable, parameter values used for the analyses shown in Tables 1 and 2 are the same as those used for the analyses illustrated by Figures 8 and 10, respectively. From the tables, we see both reliability gain and optimal duty period are sensitive to $(1-c)$. Specifically, the tables show the following:

1. A lower uncoverage $(1-c)$ favors a shorter duty period (more frequent switching) and leads to greater reliability gain, and vice versa; it is interesting to note that for the case where $\lambda = 0.0001$, the numerical results suggest that it is better to avoid switching if the uncoverage $(1-c)$ is equal to 10^{-5} or higher.

2. Reliability gain from preventive maintenance is more significant for a system with a higher failure rate (λ) only if the switching uncoverage is sufficiently low.

Table 1: Influence of Switch Coverage on $R_\theta(\phi_{\text{optimal}})$ and ϕ_{optimal}, for $\lambda = 0.0001$

$(1-c)$	$R_\theta(\theta)$	$R_\theta(\phi_{\text{optimal}})$	ϕ_{optimal}	n_{optimal}
10^{-4}	0.999952	0.999952	624.0	0
10^{-5}	0.999952	0.999952	624.0	0
10^{-6}	0.999952	0.999975	29.7	20
10^{-7}	0.999952	0.999993	15.2	40
10^{-8}	0.999952	0.999997	6.2	100
10^{-9}	0.999952	0.999999	1.9	320

5 Conclusion and Future Work

We have accomplished the initial investigation into on-board preventive maintenance for the Pluto Ex-

Table 2: Influence of Switch Coverage on $R_\theta(\phi_{\text{optimal}})$ and ϕ_{optimal}, for $\lambda = 0.0003$

$(1-c)$	$R_\theta(\theta)$	$R_\theta(\phi_{\text{optimal}})$	ϕ_{optimal}	n_{optimal}
10^{-4}	0.999587	0.999587	624.0	0
10^{-5}	0.999587	0.999757	29.7	20
10^{-6}	0.999587	0.999938	15.2	40
10^{-7}	0.999587	0.999981	6.2	100
10^{-8}	0.999587	0.999994	2.1	300
10^{-9}	0.999587	0.999998	0.65	960

press Data System. The results shown in this paper are meaningful for two reasons:

1. They illustrate that it is indeed feasible to apply analytic techniques in predicting effectiveness of and optimal duty period for on-board preventive maintenance for long-life spacecraft applications; moreover, via hierarchical model decomposition, system behavior involving deterministic transition time can be represented and evaluated in a rather simple manner.

2. They provide interesting insights regarding the effect of system failure characteristics on the effectiveness of preventive maintenance and optimal duty period. Specifically, the quantitative results reveal that switching coverage (the likelihood of a successful switching) must be sufficiently high to assure a reliability gain from the preventive-maintenance strategy discussed here.

Currently, we are in the process of elaborating the models such that some initial assumptions can be relaxed. In particular, the resulting model will allow re-initialization time during the power-on of a string to be appreciable; although by taking advantage of inherent system redundancy, re-initialization time can overlap with the duty period of the active string such that the performance overhead and its impact on the effectiveness of string switching will be minimal, it is important to study a design issue — the effect of string re-initialization time on optimal duty period. Moreover, we will consider both permanent and transient failures incurred during the power-on of a string (currently, only permanent failures are taken into account via the uncoverage $(1-c)$). Accordingly, the effectiveness of power-cycling for recovery from a transient failure will be investigated. Finally, as we observed that the length of an optimal duty period is mission-environment dependent (the influential factors include the characterization of component failure process, switching coverage, performance overhead cost and passage of time), we have been motivated to investigate the notion of adaptive on-board preventive maintenance. Specifically, we plan to conduct research on decision algorithms for adaptive duty-period optimization.

Acknowledgments

The work presented in this paper was performed for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration. The authors also gratefully acknowledge the technical support from SoHaR Incorporated in this research and in the preparation of the paper.

References

[1] Jet Propulsion Laboratory, "WBS 30000, Mission Design," Pluto Express Data System, FY95 Internal Report D-12931, Vol 2, California Institute of Technology, Pasadena, CA, Sept. 1995.

[2] Jet Propulsion Laboratory, "WBS BD000, Sciencecraft Data Subsystem," Pluto Express Data System, FY96 Internal Report, California Institute of Technology, Pasadena, CA, Sept. 1996.

[3] Y. Huang, C. Kintala, N. Kolettis, and N. D. Fulton, "Software rejuvenation: Analysis, module and applications," in *Digest of the 25th Annual International Symposium on Fault-Tolerant Computing*, (Pasadena, CA), pp. 381–390, June 1995.

[4] L. Alkalaj and M. Underwood, "Micro-electronics systems IPDT technology roadmap," Technical Report D-13276, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, Dec. 1995.

[5] L. Alkalaj, J. Klein, and M. Underwood, "The New Millennium Program microelectronics systems, advanced technology development," in *Proceedings of the 34th Aerospace Science Meeting and Exhibit*, (Reno, Nevada), Jan. 1996.

[6] K. S. Trivedi, *Probability & Statistics with Reliability, Queueing, and Computer Science Applications.* Englewood Cliffs, NJ: Prentice-Hall, 1982.

[7] S. Garg, A. Puliafito, M. Telek, and K. S. Trivedi, "Analysis of software rejuvenation using Markov regenerative stochastic Petri net," in *Proc. 6th Int'l Symposium on Software Reliability Engineering*, (Toulouse, France), pp. 180–187, Oct. 1995.

[8] S. M. Ross, *Stochastic Processes.* New York: John Wiley, 1983.