Shevasuthislip, Suntichai; Vardeman, Stephen B.

Working Paper

Development programs for one-shot systems using multiple-state design reliability models

Technical Report // Universität Dortmund, SFB 475 Komplexitätsreduktion in Multivariaten Datenstrukturen, No. 2003,43

Provided in cooperation with:
Technische Universität Dortmund

DEVELOPMENT PROGRAMS FOR ONE-SHOT SYSTEMS USING MULTIPLE-STATE DESIGN RELIABILITY MODELS

12/16/03

Suntichai Shevasuthisilp
Industrial Engineering Department
Chiang Mai University
Chiang Mai 50200, Thailand

Stephen B. Vardeman*
Statistics and IMSE Departments
Iowa State University
Ames, Iowa 50011-1210

Abstract: Design reliability at the beginning of a product development program is typically low and development costs can account for a large proportion of total product cost. We consider how to conduct development programs (series of tests and redesigns) for one-shot systems (which are destroyed at first use or during testing). In rough terms, our aim is to both achieve high final design reliability and spend as little of a fixed budget as possible on development. We employ multiple-state reliability models. Dynamic programming is used to identify a best test-and-redesign strategy and is shown to be presently computationally feasible for at least 5-state models. Our analysis is flexible enough to allow for the accelerated stress testing needed in the case of ultra-high reliability requirements, where testing otherwise provides little information on design reliability change.

Keywords: development programs, one-shot systems, multiple-state design reliability, test, redesign, optimal programs, dynamic programming, accelerated testing

1. INTRODUCTION

The purpose of this article is to identify and study the properties of optimal (test-and-redesign) development programs. We consider programs for one-shot systems such as missiles, which are destroyed at their first test or first use. Our analysis builds on those of Huang, McBeth, and Vardeman [1] (HMV1) and Moon, Vardeman, and McBeth [2] (MVM2). The previous work is generalized in two important directions, by allowing 1) multiple-state reliability modeling and 2) test failure probabilities different from normal-use failure probabilities.

Earlier work (HMV1 and MVM2) provided only crude 2-state modeling of design reliability. That modeling might be appropriate for systems with only a single potential failure mode. But the present multiple-state analysis allows for "fine" modeling of the evolution of a design (where multiple failure modes are possible and might be either eliminated or inadvertently generated by an engineering redesign).

Real test failure probabilities can be different from normal-use-condition failure probabilities in the important case of accelerated stress testing. Such testing can be necessary for sensible development programs for high-reliability systems. There, it is

* Financial support of the Deutsche Forschungsgemeinschaft (SFB 475, "Reduction of complexity in multivariate data structures") through the University of Dortmund is gratefully acknowledged.

practically impossible to obtain normal-use-condition test results that will definitively signal a change in design reliability. However, using appropriate physical acceleration factors, it can be possible to raise failure probabilities and thereby conduct tests whose results more clearly distinguish between design reliabilities.

Our interest in accelerated testing here is in this kind of increase in failure probabilities and the enabling of better empirical discrimination between reliability states. We are not concerned directly with its utility in producing quick/timely test results where normal-use testing would be prohibitively time consuming. Nor do we address the practical issues of what physical mechanisms can be employed in such acceleration or what protocols are most efficient. There is a huge literature on accelerated testing, in both statistics and engineering that address these matters. Representative work in the former can be found by beginning in Nelson [3], Meeker and Escobar [4], and Meeker and Escobar [5], and in the latter beginning in Chan [6] and Hobbs [7].

The original motivation for the line of inquiry represented here came from issues questions raised in a 1992 National Research Council (NRC) workshop on statistical issues in defense analysis and testing. A more recent report of NRC efforts in this area can be found in Cohen, Rolph, and Steffey [8]. Work similar in basic motivation to ours can be found in Gaver and Jacobs [9] and Donovan and Murphy [10]. And there is a sizeable statistical literature on the measurement of "reliability growth" (that is generally less concerned with the mechanism by which design reliability is changed than is ours) and representative work in this area is that of Fard and Dietrich [11], Erkanli, Mazzuchi, and Soyer [12], Mazzuchi and Soyer, R. [13].

2. MODELING DEVELOPMENT PROGRAMS USING MULTIPLE-STATE DESIGN RELIABILITY MODELS

We make the following basic assumptions about the process used in the development of a one-shot system.

1) An initial budget is sufficient to build N systems and all costs are in units of systems built.
2) Testing does not change design reliability. It provides information on the current design reliability state. A test result is either a "success" or a "failure" and can be purchased at a cost T. (As this test information is Bernoulli distributed, it does not typically accumulate very rapidly.)
3) Redesign has the potential to change the design reliability, but in general does not necessarily always improve it. It might degrade or improve design reliability, and can be purchased at a cost D.
4) Constants $N, T,$ and D are positive (and are not necessarily integers).
5) The effects of redesign are described by the redesign transition matrix $\mathbf{u}$ (explained more fully below) at any point in the development process where it is employed.
6) There are k possible values of design (normal-use-condition) reliability, r_i, (k design reliability states). For convenience, design reliability at a higher numbered state is greater than design reliability at a lower numbered state ($r_k > r_{k-1} > \cdots > r_1$).

7) For each reliability state, i, there is a corresponding test reliability, q_i. We will assume that test reliabilities are ordered in the same way as design reliabilities, i.e. $(q_k > q_{k-1} > \cdots > q_1)$.
8) There is no *a priori* restriction on the order in which tests and redesigns may be done in a development process. In particular, multiple tests may be made one after another (with no intervening tests) and multiple redesigns may be made one after another (with no intervening tests). (This is in contrast to the analysis of HMV1.)

2.1 The General Multiple-State Design Reliability Models

We seek a development program that produces the largest possible mean number of effective systems of a final design, given an initial budget sufficient to build N systems. We will work in units of "systems" and final (conditional) mean numbers of effective systems can be evaluated from the remaining budget at the end of development as $\lfloor B^* \rfloor \cdot \Pi^*$ (for B^* the part of the budget remaining at the end of development, $\lfloor \cdot \rfloor$ the greatest integer function, and Π^* the final design reliability, the r_i corresponding to the actual state in which development stops). Then $\mathrm{E}\left(\lfloor B^* \rfloor \cdot \Pi^*\right)$ is our objective function.

We will take what amounts to a Bayesian approach to optimization by not assuming the initial reliability state to be known exactly, but by (more generally) assuming that one has an initial/prior distribution over reliability states, $\mathbf{s}_0$. (This simple k-dimensional probability vector could be developed as an engineering "pre-test" consensus description of beliefs concerning the reliability of an initial/"0-order" design.) Then we will let $V_N(\mathbf{s})$ be the maximum of $\mathrm{E}\left(\lfloor B^* \rfloor \cdot \Pi^*\right)$ (the overall return of an optimal development plan) for the initial budget of N systems and the starting probability distribution over the states, $\mathbf{s}_0 = \mathbf{s}$.

This is a problem in (Bayesian) sequential analysis. A development process proceeds in stages. At any stage of a development program, there are 3 choices of development activity: "*test,*" "*redesign,*" and "*build.*" Each activity has a different conditional expected pay-off, which we proceed to explain in detail.

2.1.1 Testing

Testing provides information on the current design reliability state by producing a binary test result

$$X = \begin{cases} 0 & \text{if the test is passed} \\ 1 & \text{if the test is failed} \end{cases}$$

on any single test. Each test can be purchased at cost of T systems. Bayes' rule is used to update one's distribution for the current reliability state after a test is made. This, of course, requires knowledge of the vector of reliabilities of the states ($\mathbf{r}$), and a pre-test probability distribution over the states ($\mathbf{s}$). We will let

$$\mathbf{s}' = \boldsymbol{\eta}_X(\mathbf{s})$$

denote a vector specifying the updated probability distribution after testing and obtaining test outcome X. Let $\mathbf{q}$ be the vector of test reliabilities, $r(\mathbf{s})$ be the expected reliability under normal use conditions

$$r(\mathbf{s}) = \sum_{i=1}^{k} r_i s_i$$

and $q(\mathbf{s})$ be the expected reliability under test conditions,

$$q(\mathbf{s}) = \sum_{i=1}^{k} q_i s_i$$

(For case of testing under normal use conditions, $\mathbf{q} = \mathbf{r}$ and $q(\mathbf{s}) = r(\mathbf{s})$.) Then

$$\eta_{0i}(\mathbf{s}) = \frac{q_i s_i}{q(\mathbf{s})} \text{ and } \eta_{1i}(\mathbf{s}) = \frac{(1-q_i)s_i}{1-q(\mathbf{s})}$$

and the updated distribution over reliability states following a test is

$$\mathbf{s}' = \boldsymbol{\eta}_X(\mathbf{s}) = \begin{cases} (\eta_{01}(\mathbf{s}), \eta_{02}(\mathbf{s}), \ldots, \eta_{0k}(\mathbf{s})) & \text{if } X = 0 \text{ (the test is passed)} \\ (\eta_{11}(\mathbf{s}), \eta_{12}(\mathbf{s}), \ldots, \eta_{1k}(\mathbf{s})) & \text{if } X = 1 \text{ (the test is failed)} \end{cases} \quad (2.1)$$

(The forms of $\boldsymbol{\eta}_0(\mathbf{s})$ and $\boldsymbol{\eta}_1(\mathbf{s})$ in (2.1) are generalizations of ones used in HMV1.)

The remaining budget after testing will be $N - T$. Therefore, upon testing and observing X, the optimal conditional expected number of effective systems will be

$$V_{N-T}(\boldsymbol{\eta}_X(\mathbf{s}))$$

So the expected final return if a test is made is

$$q(\mathbf{s})V_{N-T}(\boldsymbol{\eta}_0(\mathbf{s})) + (1-q(\mathbf{s}))V_{N-T}(\boldsymbol{\eta}_1(\mathbf{s}))$$

2.1.2 Redesigning

We suppose that redesign is purchased at a cost of D systems lost to a final stockpile per unit of engineering effort expended in attempts to improve current design reliability. Our model allows the possibility of regressive redesigns (degrading design reliability). The (random) effects of redesigns are represented by a (stationary) Markov chain with transition matrix $\mathbf{u}$ (see the particular structures of the matrix used in our work in Appendix A.7) describing movements between design reliability states (to better or worse states) as shown in Figure 1.

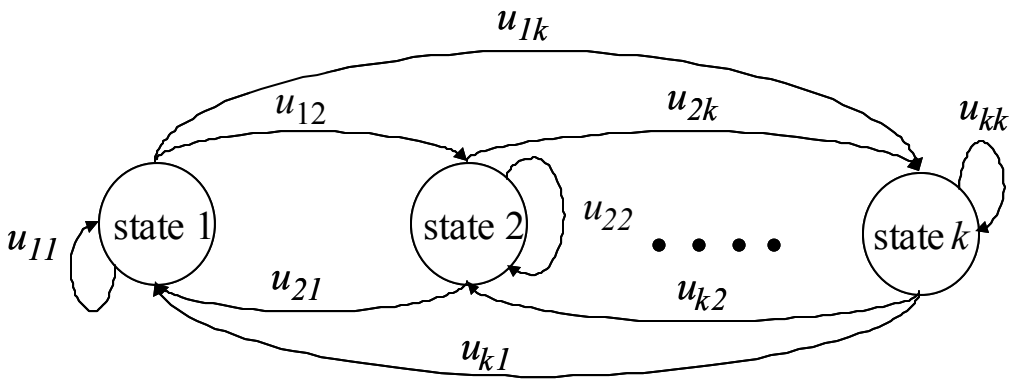

Figure 1. Possible Design Reliability Movements with Redesign in the k-State Model

Let $\boldsymbol{\delta}(\mathbf{s})$ be an updated probability distribution over the states produced from $\mathbf{s}$ by a redesign. Each

$$\delta_i(\mathbf{s}) = \sum_{j=1}^{k} u_{ji} s_j$$

or in matrix notation

$$\mathbf{s}' = \boldsymbol{\delta}(\mathbf{s}) = \left(\delta_1(\mathbf{s}), \delta_2(\mathbf{s}), \ldots, \delta_k(\mathbf{s})\right) = (s_1, s_2, ..., s_k) \cdot \begin{pmatrix} u_{11} & u_{12} & \ldots & u_{1k} \\ u_{21} & u_{22} & \ldots & u_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ u_{k1} & u_{k2} & \ldots & u_{kk} \end{pmatrix}$$

(Again, this form is a generalization of one used in MVM2.)

The remaining budget after a redesign will be $N - D$. Therefore, the optimal expected return if one redesigns is $V_{N-D}\left(\boldsymbol{\delta}(\mathbf{s})\right)$.

2.1.3 Building

The final potential development activity is "*build,*" which means that the development program is terminated and the entire remaining budget is used to build systems according to the current design and with its reliability. Therefore if one builds, the mean number of effective systems in the final stockpile is $\lfloor N \rfloor \cdot r(\mathbf{s})$.

2.1.4 Optimal Return Functions

In light of the forgoing development, the overall optimal return function is

$$V_N(\mathbf{s}) = \max\{\Psi_1, \Psi_2, \Psi_3\} \tag{2.2}$$

for

$$\Psi_1 = \lfloor N \rfloor \cdot r(\mathbf{s}),$$

$$\Psi_2 = q(\mathbf{s}) V_{N-T}\left(\boldsymbol{\eta}_0(\mathbf{s})\right) + \left(1 - q(\mathbf{s})\right) V_{N-T}\left(\boldsymbol{\eta}_1(\mathbf{s})\right)$$

and

$$\Psi_3 = V_{N-D}\left(\boldsymbol{\delta}(\mathbf{s})\right)$$

A development activity is (currently or initially) optimal at budget N and probability distribution $\mathbf{s}$ if its corresponding Ψ is maximum in display (2.2).

Optimal activities at any stage of a development program can be determined by repeatedly updating the remaining budget and probability distribution over the states, and using the optimal return function. An optimal development program will continue sequentially until "*build*" is chosen. In the case that testing and redesign both cost 1 system, the number of possible development policies could be as large as $\frac{2\cdot 3^N + 3^{N-1} - 1}{2}$. Naïve direct enumeration of all possible development policies to find a best plan would thus require that one find expected payoffs for each of a set of policies whose size grows exponentially in N.

2.2 Accelerated Stress Testing

We consider a particular model for accelerated stress testing made from the general k-state model. Our test failure probability vector is obtained by multiplying the normal-use failure probability vector $\mathbf{p} = \mathbf{1} - \mathbf{r}$ by an "acceleration factor," $a \geq 1$. That is, test reliability under accelerated testing in state i is

$$q_i = 1 - ap_i = 1 - a\left(1 - r_i\right)$$

($a = 1$ and all $q_i = r_i$ describes normal use conditions, and $1 < a < p_1^{-1}$ describes accelerated testing.)

3. SOME DIRECT CONSEQUENCES OF THE MODEL ASSUMPTIONS

Despite the move from 2-state models to k-state models and the generalization that allows $\mathbf{q} \neq \mathbf{r}$, many properties of the model specified in Section 2 carry over directly from corresponding properties of the 2-state model of MVM2. For completeness, and because several are needed in subsequent arguments, some are summarized in this section.

3.1 Properties of the Update of s After a Test

Proposition 1 The expected design reliability after testing is the same as the expected current design reliability:

$$q(\mathbf{s})\,r\left(\boldsymbol{\eta}_0(\mathbf{s})\right) + \left(1 - q(\mathbf{s})\right)r\left(\boldsymbol{\eta}_1(\mathbf{s})\right) = r(\mathbf{s})$$

This direct generalization of the 2-state Proposition 1 of MVM2 reflects the fact that testing does not change the design reliability. Testing only provides information on current design reliability.

Proposition 2 (Probability distribution over the states under an infinite sequence of tests) If one could make infinite series of tests on particular design, $\mathbf{s}$ would converge in probability to a distribution degenerate at the correct reliability state.

This statistical consistency result generalizes Proposition 2 of MVM2 and confirms that our modeling allows that with enough testing, one could with virtual certainty ascertain the true current design reliability.

Proposition 3 $\eta_{1k}(\mathbf{s}) \le s_k \le \eta_{0k}(\mathbf{s})$ and $\eta_{01}(\mathbf{s}) \le s_1 \le \eta_{11}(\mathbf{s})$.

This generalizes Proposition 3 of MVM2 and says that 1) the probability at the best design reliability state will decrease after a failed test but will increase after a successful test, and 2) the probability at the worst design reliability state will increase after a failed test but will decrease after a successful test.

3.2 Properties of the Update of $\mathbf{s}$ After a Redesign

Simple properties of stationary finite state Markov chains can be used to establish some properties of the effects of redesign generalizing the 2-state Propositions 4 and 5 of MVM2.

Proposition 4 (Probability distribution over the states after a single redesign)

Case 1 ($u_{ij} > 0$ for $i \le j$ and $u_{ij} = 0$ for $i > j$): For any i, $\sum_{j=1}^{i} \delta_j(\mathbf{s}) \le \sum_{j=1}^{i} s_j$.

Case 2 ($u_{ii} = 1$ for all $i = 1, 2, \ldots, k$): $\boldsymbol{\delta}(\mathbf{s}) = \mathbf{s}$; Redesign has no effect on $\mathbf{s}$.

We will call the situation of Case 1 that of "non-regressive redesigns" following MVM2. In Case 1, Proposition 4 says that the distributions $\boldsymbol{\delta}(\mathbf{s})$ and $\mathbf{s}$ are stochastically ordered (and so, for example, $\delta_k(\mathbf{s}) \ge s_k$ and $\delta_1(\mathbf{s}) \le s_1$).

Proposition 5 (Probability distribution over the states under an infinite sequence of redesigns)

Case 1 The Markov chain transition matrix $\mathbf{u}$ is irreducible, positive recurrent, and aperiodic: Under an infinite sequence of redesigns $\mathbf{s}$ converges to a steady-state probability vector $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_k)$ that may be computed by solving the linear equations $\boldsymbol{\beta} = \boldsymbol{\beta} \cdot \mathbf{u}$ and $\beta_1 + \beta_2 + \cdots + \beta_k = 1$.

Case 2 ($u_{ij} > 0$ for $i \le j$ and $u_{ij} = 0$ for $i > j$): Under an infinite sequence of redesigns s_k converges to 1 and expected design reliability $r(\mathbf{s})$ converges to r_k.

The Case 1 result promises a kind of "law of diminishing returns" for possibly regressive redesigns and suggests that in many contexts (at least in the absence of testing

adequate to definitively indicate a true current reliability state) relatively few redesigns may be appropriate in a program. On the other hand, the Case 2 result says that if one makes an infinite sequence of non-regressive redesigns, eventually design reliability will be at the best design reliability state.

4. ANALYSIS OF OPTIMAL DEVELOPMENT PLANS

The properties of our model recorded in Section 3 are important, but don't immediately concern optimal development plans. In this section we first state those properties of optimal development that allow their computation, and then describe how we use those properties and simulations in our analysis of plan behavior.

4.1 Optimal Next Actions and Evaluating $V_N(\mathbf{s})$

Proposition 6 If $N < 1 + D$, stopping is an optimal next action and $V_N(\mathbf{s}) = \lfloor N \rfloor \cdot r(\mathbf{s})$.

This proposition says that redesign or testing will not be beneficial if the remaining budget after making a redesign ($N - D$) would be below that required to build one system. This guarantees that at least one system of the final design will be built.

Proposition 7 If $N < 1 + T + D$, only stopping and redesign are potentially optimal next actions and

$$V_N(\mathbf{s}) = \max_{l \text{ s.t. } N - lD \geq 1} \left\{ \lfloor N - lD \rfloor \cdot r\left(\boldsymbol{\delta}^l(\mathbf{s})\right) \right\}$$

(for $\boldsymbol{\delta}^l(\mathbf{s})$ the l-fold composition of $\boldsymbol{\delta}$ with itself).

This proposition says that testing is not beneficial if the remaining budget after making a test would not be sufficient to purchase at least one redesign and build one system. It is better to stop or do a number of redesigns that produces the maximum expected payoff.

The following is simply a formalization of display (2.2) and says that for large current budgets, stopping, testing and redesign are all potential next actions.

Proposition 8 If $N \geq 1 + T + D$, stopping, testing, and redesign are potentially optimal next actions and

$$V_N(\mathbf{s}) = \max\left\{ \lfloor N \rfloor \cdot r(\mathbf{s}), q(\mathbf{s}) V_{N-T}\left(\boldsymbol{\eta}_0(\mathbf{s})\right) + \left(1 - q(\mathbf{s})\right) V_{N-T}\left(\boldsymbol{\eta}_1(\mathbf{s})\right), V_{N-D}\left(\boldsymbol{\delta}(\mathbf{s})\right) \right\}$$

4.2 Analysis of Optimal Development Programs

Our analysis of optimal development programs consists of two steps. First, using the results of Section 4.1 we compute and store $V_m(\mathbf{s})$ at each possible remaining budget point m, over a grid of probability distributions for the states (a grid of $\mathbf{s}$ vectors in the simplex of probability vectors in k-space). Second, we investigate the behavior of the

optimal plans using simulation. During the simulation process, an optimal activity at any current budget N_c and current probability vector for the states $\mathbf{s}_c$ is determined using the information stored in the first step.

4.2.1 Computation of the Optimal Plans and Expected Payoff for an Initial Budget of N

The computation of optimal returns $V_m(\mathbf{s})$ at each possible remaining budget point (m) for all possible probability distributions over the states ($\mathbf{s}$) proceeds by "backwards induction." This process moves from the smallest to the largest possible remaining budget point. Inputs are the redesign transition matrix $\mathbf{u}$, the initial budget N, the test cost T, the redesign cost D, the design reliability vector $\mathbf{r}$, and parameters for a (k, M)-simplex-lattice design specifying the grid of vectors $\mathbf{s}$ over which optimal payoffs will be evaluated

In our first step we:

a) determine all possible remaining budget points, m, that might be reached in the development process using

$$m = N - k_1 T - k_2 D \geq 1$$

where k_1 and k_2 (respectively a number of tests and a number of redesigns in the sequence) are nonnegative integers,

b) sort the possible remaining budget points in ascending order

$$1 \leq m_1 < m_2 < \ldots < m_b = N$$

where b is the total number of possible remaining budget points,

c) recursively determine optimal returns $V_m(\mathbf{s})$ for all $\mathbf{s}$ on a grid by applying Propositions 6-8 (and interpolations where needed), starting from m_1 and proceeding to $m_b = N$. In this process:

1) the probability distributions ($\mathbf{s}$) over the states are elements of a (k, M)-simplex-lattice design. So each component of $\mathbf{s}$ is a multiple of M^{-1}, and $s_1 + s_2 + \cdots + s_k = 1$. (We used $M = 5{,}000$ in our analyses.)

2) the procedure for determining optimal returns $V_m(\mathbf{s})$ is:

$$V_m(\mathbf{s}) = \begin{cases} \Psi_1 & \text{if } 1 \leq m < 1 + D \\ \max\{\Psi_1, \Psi_2\} & \text{if } 1 + D \leq m < 1 + T + D \\ \max\{\Psi_1, \Psi_2, \Psi_3\} & \text{if } 1 + T + D \leq m \end{cases}$$

where

$\Psi_1 = \lfloor m \rfloor \cdot r(\mathbf{s})$,

$\Psi_2 = q(\mathbf{s})\,\text{INTERP}\left[V_{m-T}(\boldsymbol{\eta}_0(\mathbf{s}))\right] + (1 - q(\mathbf{s}))\,\text{INTERP}\left[V_{m-T}(\boldsymbol{\eta}_1(\mathbf{s}))\right]$, and

$\Psi_3 = \text{INTERP}\left[V_{m-D}(\boldsymbol{\delta}(\mathbf{s}))\right]$

and the interpolation method INTERP[ ] is described in Appendix A.6.

Some discussion is in order concerning the magnitude of the computational burden implied by a) through c) above. In the first place, we note that when T is an integer multiple of D (or vice versa) b is linear in N. In a worst case, where no integer multiple of T is an integer multiple of D, b is only quadratic in N (being approximately $\frac{1}{2}\left(\frac{N}{T}+1\right)\left(\frac{N}{D}+1\right)$). So (unlike what happens when one tries to directly enumerate all possible plans) the computational burden that must be faced to determine optimal policies doesn't grow rapidly in the budget size.

Further, the number of points in a (k, M)-simplex-lattice design is exactly $(k+M-1)!/M!(k-1)!$ This means that for fixed k, the computational burden grows with M at order M^{k-1}. M controls the precision with which the functions $V_m(\mathbf{s})$ are computed (and therefore, the precision with which an optimal policy is identified). For the small values of k we used, this suggests an attractive relationship between precision and computational time. And the count of design points also shows that for fixed M, the computational burden grows with k at order $(k-1)^M$. While this is strictly speaking "polynomial" order, for the kind of large M we've employed (to get good precision), this grows very fast with k. And from a practical point of view, it was the number of states, k, that was the limiting factor in our computations.

4.2.2 Simulating the Behavior of an Optimal Development Plan

We studied the expected (under a particular prior/initial probability distribution $\mathbf{s}_0$) behavior of optimal (for that $\mathbf{s}_0$) development plans using simulation. Simulation of the development plan for an initial probability distribution, $\mathbf{s}_0$, and an initial budget of N involves randomly generating test results and the effects of redesigns. (Test results are Bernoulli distributed according to the current reliability state, and that state evolves randomly according to the transition matrix.) During the simulation process, an optimal next activity at any point is determined by consulting the optimal returns computed and stored as described in Section 4.2.1. One simulation "trial" then consists of beginning at a) below and ending at d):

a) An initial reliability state is generated according to the initial distribution over the states ($\mathbf{s}_0$).
b) Starting at initial budget N and initial probability vector $\mathbf{s}_0$, an optimal next activity at any current budget N_c and current probability vector for states $\mathbf{s}_c$ is determined by using Propositions 6-8 and the stored values of $V_m(\mathbf{s})$.
c) Depending upon what activity is prescribed in b), the current budget N_c, the probability vector $\mathbf{s}_c$, and the current reliability state are updated as follows:

1) If the activity prescribed is "redesign," the budget is reduced to $N' = N_c - D$, the probability vector is updated to $\mathbf{s}' = \boldsymbol{\delta}(\mathbf{s}_c)$, and a new real reliability state is generated from the current one using the distribution specified by the appropriate row of the transition matrix $\mathbf{u}$.
2) If the activity prescribed is "test," the budget is reduced to $N' = N_c - T$, a test result X is generated by using the current real design reliability state, the probability vector is updated to $\mathbf{s}' = \boldsymbol{\eta}_X(\mathbf{s}_c)$, and the real design reliability state is not changed.

d) The development process is terminated when "build" becomes an optimal next activity, and we record the conditional expected number of effective systems built $\lfloor B^* \rfloor \cdot \Pi^*$, for B^* the final remaining budget and Π^* the final realized design reliability.

Such trials (beginning at a) and ending at d)) are made until a desired number has been accumulated.

5. SOME NUMERICAL RESULTS

5.1 Relationship Between Computing Time and Number of Design Reliability States

Table 5.1 shows some average computing times for 3-state, 4-state, and 5-state models. The computing time has two parts. First is the set-up time needed to build a table of optimal returns for all possible probability distributions at all possible remaining budget points. The set-up time for given k and N is approximately constant in the other problem parameters and mostly depends on the number of $\mathbf{s}$ grid points used (namely $(k+M-1)!/M!(k-1)!$). Second, there is an average simulation time used to study a development plan. The average (across initial distributions ($\mathbf{s}_0$)) simulation times for all 25,000 trials of the development programs for 3, 4, and 5-state models are displayed (for 66, 56, and 70 initial distributions $\mathbf{s}_0$ respectively). As we expect, the computing time increases rapidly in the number of design reliability states.

Table 5.1: Computing Time for $N = 1000,\ T = D = 5$, and $(r_1, r_k) = (0.10, 0.90)$ (the $r_{i+1} - r_i$ constant for a given k)

	Computing Time (Minutes)*		
	3-State Model	4-State Model	5-State Model
Set-up Time	39	314	2,058
Average Simulation Time for Problem 1 (using $\mathbf{u}_a$)	0.197	1.697	9.685
Average Simulation Time for Problem 2 (using $\mathbf{u}_b$)	0.061	0.411	2.871

*(Using an 800 MHz Pentium II computer with 512 MB RAM, running programs developed in C++)

5.2 k-State Results Without Accelerated Testing (the Effects of Model Parameters on Optimal Plan Behavior)

We consider how the test cost (T), the redesign cost (D), the redesign transition matrix ($\mathbf{u}$), and the design reliability vector ($\mathbf{r}$) affect the behavior and performance of optimal plans. Most of the discussion here is based primarily on extensive simulation results (for a total of 7,128 different problems) using 3-state reliability models with $\mathbf{q} = \mathbf{r}$ reported in Shevasuthisilp [14]. We have also done some simulations for 4- and 5-state models to verify that our methods and analyses are capable of handling larger numbers of states and produce qualitatively the same results as for 3-state models.

Some representative results from the large set reported in Shevasuthisilp [14] are summarized in tables in Appendix A.8. Model parameters and plan characteristics recorded there are: initial probability distribution for the states ($\mathbf{s}_0$) and average probability distribution at program end ($\overline{\mathbf{s}^*}$), initial expected reliability ($r(\mathbf{s}_0)$) and average reliability at program end ($\overline{r(\mathbf{s}^*)}$), the expected number of effective systems without ($\lfloor N \rfloor \cdot r(\mathbf{s}_0)$) and with ($V_N(\mathbf{s}_0)$) development, the optimal first action (F, where $F = 1$ is "build", $F = 2$ is "test", and $F = 3$ is "redesign"), and the average numbers of systems built ($\overline{\lfloor B^* \rfloor}$), redesigns made ($\overline{\Delta^*}$), and tests made ($\overline{\mathrm{T}^*}$). The averages in the tables come from 25,000 simulation trials per case and have very small standard errors. (The sizes of these standard errors are summarized in the final table of Appendix A.8.) The particular forms of $\mathbf{u}$ referred to in the tables and their captions are given and discussed briefly in Appendix A.7.

Table A.1 illustrates how the redesign cost (D) affects the behavior of optimal plans. The test cost is fixed at $T = 5$ and redesign costs are $D = 5$ and 50. We find that

as the redesign cost increases, the average number of redesigns made by optimal plans, the optimal mean number of effective systems built, and the average final expected reliability all decrease. The findings are sensible, because when the cost of redesign is high, it is not economical to do many redesigns. Increasing the redesign cost also affects the number of tests made and the optimal first activity in a development program. As the redesign cost increases, the optimal first activity can change from test to build. It is not beneficial to do testing alone without following poor test results with redesign.

Table A.2 illustrates how the test cost (T) affects the behavior of optimal plans. The redesign cost is fixed at $D = 5$ and test costs are $T = 5$ and 50. We find that as the test cost increases, the average number of tests made by optimal plans, the optimal mean number of effective systems, and the average final expected reliability all decrease. The findings are reasonable. An optimal plan uses few tests when testing is expensive. In such cases there is little empirical reliability information available for design improvement. It is also evident from Table A.2 that the optimal first activity in a development program tends to change from test to build as test cost increases.

Table A.3 illustrates how the redesign transition matrix ($\mathbf{u}$) affects the behavior of optimal plans. Redesign and test costs are fixed at $T = D = 5$. We find that when a more effective redesign transition matrix is available ($\mathbf{u}_b$ in place of $\mathbf{u}_a$) the average number of redesigns made and amount of resources devoted to development ($N - \overline{\lfloor B^* \rfloor}$) by optimal plans decrease, but the mean number of effective systems increases. Moving to a better redesign transition matrix also decreases the number of tests made and tends to change the optimal first activity from test to redesign.

Table A.4 enables comparisons of optimal plans for two different design reliability vectors ($\mathbf{q} = \mathbf{r} = (0.10,0.30,0.50)$ and $(0.80,0.85,0.90)$). We find that optimal programs for low design reliability problems employ more tests and redesigns than optimal programs for high design reliability problems. This agrees with intuition that when reliability is low, more development resources should be devoted to improve current design reliability.

Table A.5 summarizes simulation results for high reliability problems ($\mathbf{q} = \mathbf{r} = (0.80, 0.85, 0.90)$) with redesign transition matrix $\mathbf{u}_b$, $T = 5$, and $D = 5$ and 50. The nature of optimal plans is "unusual" for these cases, in that no test is made in any optimal program. This was consistent across all such high design reliability cases we studied (495 combinations of parameters using T and D of 5,10 and 50, 4 redesign transition matrices, and 66 initial probability distributions). Since design reliability is always high (always at least 0.8), the likelihood of test failure in any reliability state is very low, and testing does not produce much useful information. Testing only wastes limited development resources if it does not provide a basis to discriminate effectively among reliability states.

In all, careful examination of our simulation results confirms that our mathematics is behaving qualitatively "exactly as it should," in complete accord with intuition. What it provides is, of course, exact quantitative guidelines consistent with any set of input model parameters.

5.3 k -State Results with Accelerated Testing

In general, the acceleration of testing described in Section 2.2 does affect the behavior of optimal development programs in high reliability problems. The effects are positive, since the acceleration provides more useful reliability information for guiding design improvement. In our high reliability cases, expected returns under accelerated testing were always higher than expected returns under the normal use testing conditions. We also found that in the cases we studied, the larger the acceleration factor, the stronger the positive effects. The optimal plans under acceleration are often intuitively more appealing than without acceleration. By using acceleration, optimal plans often change from employing only redesign(s) or immediate build, to using a mixed sequence of tests and redesigns before building.

Tables A.7 and A.8 (for $\mathbf{r} = (0.80, 0.85, 0.90)$) and Table A.9 (for $\mathbf{r} = (0.900, 0.945, 0.990)$) of the Appendix A.9 summarize numbers of cases (at 66 initial probability distributions $\mathbf{s}_0$ on an $M = 10$ grid) affected by acceleration of testing. If optimal plans are affected (changed by the use of $a > 1$), the differences (I) between expected optimal returns under accelerated testing and normal use conditions testing are summarized. (As in our other simulations, 25,000 runs were made for each case.)

Table A.7 illustrates how the behavior of optimal plans is affected by using the maximum possible acceleration factor at different levels of redesign cost. The number of optimal plans affected decreases as the redesign cost (D) increases. This suggests that accelerated testing is less effective as the redesign cost increases. More informative testing alone is not beneficial if redesign cannot be economically made after testing. When the redesign cost is high, one is pushed towards an initial "build" action.

Table A.8 illustrates how the behavior of optimal plans is affected by using the maximum possible acceleration factor for three different redesign transition matrices. Most of the optimal plans are affected when the redesign transition matrix is $\mathbf{u}_\mathrm{b}$ ($g = 0.05, f = 0.75$), a redesign transition matrix describing moderately effective redesigns. Acceleration has almost no effect when redesigns are highly effective (for the matrix $\mathbf{u}_\mathrm{c}$ ($g = 0.05, f = 1.00$) redesign always improves system reliability). When redesign is highly effective, this fact always dominates the effect of (even accelerated) testing. Accelerated testing has less effect on the optimal plans for the least effective redesign mechanism $\mathbf{u}_\mathrm{a}$ ($g = 0.05, f = 0.25$) than for $\mathbf{u}_\mathrm{b}$ ($g = 0.05, f = 0.75$). For this case, redesign is rarely an optimal activity and (accelerated) testing is not called for either, since testing alone is not beneficial if it is not followed by effective redesign.

Table A.9 shows how the behavior of optimal plans is affected by the acceleration factor (a). The number of optimal plans affected and performance measures increase as the acceleration factor increases. This is intuitively appealing because testing under high acceleration provides more informative indications of design reliability.

6. FINAL COMMENTS

This paper's extension of the earlier analyses of HMV1 and MVM2 allows for far more realistic modeling of a development process. Of course, using more than two reliability states creates the burden of specifying more detailed vectors $\mathbf{r}$ and $\mathbf{s}_0$ and a defensible form for $\mathbf{u}$, and use of the $\mathbf{q} \neq \mathbf{r}$ idea requires specification of a sensible/physically appropriate link between $\mathbf{r}$ and $\mathbf{q}$. (We have used one synthetic choice, others potentially suggested by physical theory in particular contexts are clearly possible.) But this situation is as it always is when contemplating the use of nested mathematical models. A balance must be struck between increasing potential model fidelity given appropriate values for increasing numbers of parameters, and one's diminishing ability to adequately specify them. Our work demonstrates that computationally at least, there is no real problem in moving to substantially increased model complexity.

Should one wish to improve modeling flexibility and potential fidelity yet further, there a several directions that more complex modeling could take. Among them are at least the following:

a) We have used a simple stationary Markov chain transition mechanism and single cost to describe redesign. But this could be generalized. Different kinds of redesigns (potentially only available at different stages of a program) with different costs and different likely efficacies might be considered. For example, designers could gain experience over time, or it might be very difficult to redesign effectively late in a development program. So, the effects of redesign might be described using matrices that change with the number of redesigns made or the remaining budget. Or it is possible and potentially more realistic to describe redesign cost as a function of remaining budget, the type of redesign transition matrix applied, or the current distribution over design reliability states.

b) There are varieties of ways in which the modeling of testing might be made more flexible/general. Several different kinds of tests with different costs and different test failure probabilities might be considered. Or, where there is test data that stands behind pass-fail outcomes, incorporating parametric modeling of those results could lead to more effective development programs. (For example, a normal variable might cause a test failure if it exceeds some physically important limit. Using the value of the variable in modeling rather than simply the information as to whether it exceeds the test limit can improve expected payoff. See Huang [15] and Senoglu and Vardeman [16] in this regard for 2-state problems.)

A. APPENDIX

In this appendix we provide proofs for several of the propositions. (We present only those proofs that are both fundamentally different from any presented in MVM2 for 2-state cases and also perhaps not completely obvious.) We then give some details for

the interpolation method we used in our computations and specify the redesign transition matrices we employed in our numerical work. Finally, tables summarizing some of our numerical results are presented.

A.1 Proof of Proposition 1

Note that

$$q(\mathbf{s})r\left(\boldsymbol{\eta}_0(\mathbf{s})\right)+\left(1-q(\mathbf{s})\right)r\left(\boldsymbol{\eta}_1(\mathbf{s})\right)=q(\mathbf{s})\sum r_i\eta_{0i}(\mathbf{s})+\left(1-q(\mathbf{s})\right)\sum r_i\eta_{1i}(\mathbf{s})$$

$$=\sum r_i q_i s_i+\sum r_i\left(1-q_i\right)s_i$$

$$=r(\mathbf{s})$$

□

A.2 Proof of Proposition 3

Consider showing that $\eta_{1k}(\mathbf{s})\le s_k$. Note that

$$s_k-\eta_{1k}(\mathbf{s})=s_k-\frac{s_k(1-r_k)}{1-(r_1s_1+r_2s_2+\cdots+r_ks_k)}$$

$$=\frac{s_k[r_k(1-s_k)-r_1s_1-r_2s_2-\cdots-r_{k-1}s_{k-1}]}{1-r(\mathbf{s})}$$

$$=\frac{s_k[r_k(s_1+s_2+\cdots+s_{k-1})-r_1s_1-r_2s_2-\cdots-r_{k-1}s_{k-1}]}{1-r(\mathbf{s})}$$

$$=\frac{s_k[(s_1(r_k-r_1)+s_2(r_k-r_2)+\cdots+s_{k-1}(r_k-r_{k-1})]}{1-r(\mathbf{s})}$$

Since $r_k>r_{k-1}>\cdots>r_1$, the numerator above is nonnegative.

Arguments for the inequalities $\eta_{0k}(\mathbf{s})\ge s_k,\eta_{01}(\mathbf{s})\le s_1,$ and $s_1\le\eta_{11}(\mathbf{s})$ are analogous. □

A.3 Proof of Proposition 4 (Case I)

For the non-regressive case,

$$\sum_{j=1}^{i}\delta_j(\mathbf{s})=\sum_{j=1}^{i}\left(\sum_{l=1}^{j}s_lu_{lj}\right)=\sum_{l=1}^{i}\sum_{j=l}^{i}s_lu_{lj}=\sum_{l=1}^{i}s_l\sum_{j=1}^{i}u_{lj}\le\sum_{j=1}^{i}s_j \quad \square$$

A.4 Proof of Proposition 6

First suppose that $T \geq D$. Here $N < 1+T$, so making either a test or redesign will reduce the current budget below that required to build at least one system of the final design. Therefore stopping is an optimal next action and $V_N(\mathbf{s}) = \lfloor N \rfloor \cdot r(\mathbf{s})$.

For $T < D$, consider first the case where $N < 1+D$ and $N < 1+T$. Making either a redesign or a test will reduce the budget below that required to build at least one system of the final design. Therefore stopping is an optimal next action.

Next consider the situation where $N < 1+D$ and $1+T \leq N < 1+2T$. Stopping and making a test are potentially optimal next actions. So

$$\begin{aligned} V_N(\mathbf{s}) &= \max\left\{\lfloor N \rfloor \cdot r(\mathbf{s}), q(\mathbf{s}) V_{N-T}(\boldsymbol{\eta}_0(\mathbf{s})) + (1-q(\mathbf{s})) V_{N-T}(\boldsymbol{\eta}_1(\mathbf{s}))\right\} \\ &= \max\left\{\lfloor N \rfloor \cdot r(\mathbf{s}), q(\mathbf{s})\left(\lfloor N-T \rfloor \cdot r(\boldsymbol{\eta}_0(\mathbf{s}))\right) + (1-q(\mathbf{s}))\left(\lfloor N-T \rfloor \cdot r(\boldsymbol{\eta}_1(\mathbf{s}))\right)\right\} \\ &= \max\left\{\lfloor N \rfloor \cdot r(\mathbf{s}), \lfloor N-T \rfloor \cdot \left(q(\mathbf{s}) r(\boldsymbol{\eta}_0(\mathbf{s})) + (1-q(\mathbf{s})) r(\boldsymbol{\eta}_1(\mathbf{s}))\right)\right\} \end{aligned}$$

Apply Proposition 1 and this becomes

$$V_N(\mathbf{s}) = \max\left\{\lfloor N \rfloor \cdot r(\mathbf{s}), \lfloor N-T \rfloor \cdot r(\mathbf{s})\right\} = \lfloor N \rfloor \cdot r(\mathbf{s})$$

$\vdots$

Finally, (for $T < D$) suppose that $N < 1+D$ and $1+(l-1)T \leq N < 1+lT$ for a positive integer l. The optimal expected payoffs can be determined by induction and again applying Proposition 1, and are found to be $\lfloor N \rfloor \cdot r(\mathbf{s})$. Since this expected payoff is available by stopping, we may therefore also conclude that stopping is an optimal next action. □

A.5 Proof of Proposition 7

For $T \leq D$ the potential optimal next options are stopping, redesign, and testing, and the optimal return is from (2.2)

$$V_N(\mathbf{s}) = \max\left\{\lfloor N \rfloor \cdot r(\mathbf{s}), q(\mathbf{s}) V_{N-T}(\boldsymbol{\eta}_0(\mathbf{s})) + (1-q(\mathbf{s})) V_{N-T}(\boldsymbol{\eta}_1(\mathbf{s})), V_{N-D}(\boldsymbol{\delta}(\mathbf{s}))\right\}$$

An optimal next action after making a test is stopping (using Proposition 6, since the remaining budget after making a test is less than $1+D$). Thus

$$\begin{aligned} V_N(\mathbf{s}) &= \max\left\{\lfloor N \rfloor \cdot r(\mathbf{s}), q(\mathbf{s}) \lfloor N-T \rfloor \cdot r(\boldsymbol{\eta}_0(\mathbf{s})) + (1-q(\mathbf{s})) \lfloor N-T \rfloor \cdot r(\boldsymbol{\eta}_1(\mathbf{s})), \lfloor N-D \rfloor \cdot r(\boldsymbol{\delta}(\mathbf{s}))\right\} \\ &= \max\left\{\lfloor N \rfloor \cdot r(\mathbf{s}), \lfloor N-T \rfloor \cdot \left(q(\mathbf{s}) r(\boldsymbol{\eta}_0(\mathbf{s})) + (1-q(\mathbf{s})) r(\boldsymbol{\eta}_1(\mathbf{s}))\right), \lfloor N-D \rfloor \cdot r(\boldsymbol{\delta}(\mathbf{s}))\right\} \end{aligned}$$

Apply Proposition 1 to the second term in braces. The recursion (2.2) becomes

$$V_N(\mathbf{s}) = \max\left\{\lfloor N \rfloor \cdot r(\mathbf{s}), \lfloor N-T \rfloor \cdot r(\mathbf{s}), \lfloor N-D \rfloor \cdot r(\boldsymbol{\delta}(\mathbf{s}))\right\}$$

The first term is greater than the second term, so in fact

$$V_N(\mathbf{s}) = \max\left\{\lfloor N \rfloor \cdot r(\mathbf{s}), \lfloor N-D \rfloor \cdot r(\boldsymbol{\delta}(\mathbf{s}))\right\}$$

and potentially optimal next actions are stopping and redesign.

For $T > D$, consider first the case where $N < 1+T+D$ and $1+D \le N < 1+2D$. By the same argument as used in the $T \le D$ case, testing is not an option. Thus, potential actions are stopping and redesign and the expected payoffs are $\lfloor N \rfloor \cdot r(\mathbf{s})$ and $V_{N-D}(\boldsymbol{\delta}(\mathbf{s}))$ respectively. Then apply Proposition 6 to the second term and it becomes $\lfloor N-D \rfloor \cdot r(\boldsymbol{\delta}(\mathbf{s}))$, so here

$$V_n(\mathbf{s}) = \max_{l=0,1} \left\{ \lfloor N - lD \rfloor \cdot r\left(\boldsymbol{\delta}^l(\mathbf{s})\right) \right\}$$

$\vdots$

Finally, (for $T > D$) consider the case where $N < 1+T+D$ and $1+(l-1)D \le N < 1+lD$ for a positive integer l. Again testing is not an option, potential next actions are stopping and redesign, and the expected payoffs are

$$V_N(\mathbf{s}) = \max\left\{ \lfloor N \rfloor \cdot r(\mathbf{s}), V_{N-D}(\boldsymbol{\delta}(\mathbf{s})) \right\}$$

By induction

$$V_{N-D}(\boldsymbol{\delta}(\mathbf{s})) = \max_{0 \le m \le l-1} \left\{ \lfloor (N-D) - mD \rfloor \cdot r\left(\boldsymbol{\delta}^m(\boldsymbol{\delta}(\mathbf{s}))\right) \right\}$$

So

$$V_N(\mathbf{s}) = \max_{l \text{ s.t. } N-lD \ge 1} \left\{ \lfloor N - lD \rfloor \cdot r\left(\boldsymbol{\delta}^l(\mathbf{s})\right) \right\}$$

and the possible options are stopping or doing at most $\left\lfloor \frac{N-1}{D} \right\rfloor$ redesigns. □

A.6 The Interpolation Method

Interpolation is needed during the process of recursively determining expected returns $V_m(\mathbf{s})$, where at budget m and probability vector $\underline{s}$ an updated probability distribution $\mathbf{s}'$ does not match exactly any point on the available grid of probability vectors. We use multidimensional linear interpolation (and can hope that it will often be very accurate in our application, since exactly as indicated in Proposition 11 of MVM2 for the 2-state model, every $V_m(\mathbf{s})$ is piecewise linear in $\mathbf{s}$). The following is a complete description of our method for the $k = 3$ case. (Details for larger *k* are similar.)

Let $V_m(\mathbf{s})$ be a value to be interpolated. Write $\mathbf{s} = (s_1, s_2, s_3)$ and think of $V_m(\mathbf{s})$ as a function of s_2 and s_3, and as already evaluated for those $\mathbf{s}$ whose entries are multiples of M^{-1} for a positive integer M. Let

$$w_2 = Ms_2 \text{ and } w_3 = Ms_3$$

Define both w_2^- and w_2^+ as w_2 if w_2 is an integer, and as the two consecutive integers with $w_2^- < w_2 < w_2^+$ otherwise. Similarly define w_3^- and w_3^+.

Then, if $w_2^+ + w_3^+ \le M$ define for $i = 2, 3$

$$c_i = \begin{cases} 0 & \text{if } w_i^- = w_i^+ \\ \dfrac{w_i - w_i^-}{w_i^+ - w_i^-} & \text{otherwise} \end{cases}$$

and set

$$\text{INTERP}\left[V_m\left(s_2, s_3\right)\right] = (1-c_2)(1-c_3)V_m(\frac{w_2^-}{M}, \frac{w_3^-}{M}) + c_2(1-c_3)V_m(\frac{w_2^+}{M}, \frac{w_3^-}{M})$$
$$+ (1-c_2)c_3V_n(\frac{w_2^-}{M}, \frac{w_3^+}{M}) + c_2c_3V_m(\frac{w_2^+}{M}, \frac{w_3^+}{M})$$

If, on the other hand, $w_2^+ + w_3^+ > M$, set

$$\text{INTERP}\left[V_m\left(s_2, s_3\right)\right] = \left(1-(w_2 - w_2^-) - (w_3 - w_3^-)\right)V_m(\frac{w_2^-}{M}, \frac{w_3^-}{M}) + (w_2 - w_2^-)V_m(\frac{w_2^+}{M}, \frac{w_3^-}{M})$$
$$+ (w_3 - w_3^-)V_n(\frac{w_2^-}{M}, \frac{w_3^+}{M})$$

A. 7 Transition Matrices Describing Effects of Redesigns

We parameterized the transition matrices **u** used in our simulations in terms of 1) a diagonal probability, g , representing the likelihood of affecting no reliability change through redesign and 2) a conditional probability of improving design reliability given a reliability change, f . Two "non-regressive" ($f = 1$) and two other transition matrices allowing the possibility of design degradation ($f < 1$) were created using $g = 0.05$ and $f = 0.25, 0.75$, and using $g = 0.05, 0.50$ and $f = 1.00$ respectively in the following forms. For $k = 3$ we used

$$\begin{pmatrix} g + (1-f)(1-g) & \dfrac{f(1-g)}{2} & \dfrac{f(1-g)}{2} \\ (1-f)(1-g) & g & f(1-g) \\ \dfrac{(1-f)(1-g)}{2} & \dfrac{(1-f)(1-g)}{2} & g + f(1-g) \end{pmatrix}$$

For $k = 4$ we used

$$\begin{pmatrix} g + (1-f)(1-g) & \dfrac{f(1-g)}{3} & \dfrac{f(1-g)}{3} & \dfrac{f(1-g)}{3} \\ (1-f)(1-g) & g & \dfrac{f(1-g)}{2} & \dfrac{f(1-g)}{2} \\ \dfrac{(1-f)(1-g)}{2} & \dfrac{(1-f)(1-g)}{2} & g & f(1-g) \\ \dfrac{(1-f)(1-g)}{3} & \dfrac{(1-f)(1-g)}{3} & \dfrac{(1-f)(1-g)}{3} & g + f(1-g) \end{pmatrix}$$

And for $k = 5$ we used

$$\begin{pmatrix} g+(1-f)(1-g) & \frac{f(1-g)}{4} & \frac{f(1-g)}{4} & \frac{f(1-g)}{4} & \frac{f(1-g)}{4} \\ (1-f)(1-g) & g & \frac{f(1-g)}{3} & \frac{f(1-g)}{3} & \frac{f(1-g)}{3} \\ \frac{(1-f)(1-g)}{2} & \frac{(1-f)(1-g)}{2} & g & \frac{f(1-g)}{2} & \frac{f(1-g)}{2} \\ \frac{(1-f)(1-g)}{3} & \frac{(1-f)(1-g)}{3} & \frac{(1-f)(1-g)}{3} & g & \frac{f(1-g)}{1} \\ \frac{(1-f)(1-g)}{4} & \frac{(1-f)(1-g)}{4} & \frac{(1-f)(1-g)}{4} & \frac{(1-f)(1-g)}{4} & g+f(1-g) \end{pmatrix}$$

A.8 Tables Summarizing Some Numerical Results Without Acceleration of Testing

Table A.1: Simulation Results for $N = 1{,}000$, $T = 5$, $\mathbf{u}_a$ ($g = 0.05, f = 0.25$), and $\mathbf{q} = \mathbf{r} = (0.10, 0.50, 0.90)$

$\mathbf{s}_0$	D	$\overline{\mathbf{s}^*}$	$r(\mathbf{s}_0)$	$\overline{r(\mathbf{s}^*)}$	$\lfloor N \rfloor \cdot r(\mathbf{s}_0)$	$V_N(\mathbf{s}_0)$	F	$\overline{\lfloor B^* \rfloor}$	$\overline{\Delta^*}$	$\overline{\mathrm{T}^*}$
(1.00,0.00,0.00)	5	(0.000,0.055,0.945)	0.100	0.878	100	770.31	3	876.71	8.62	16.02
	50	(0.072,0.230,0.698)		0.751		515.52	3	652.19	5.93	10.22
(0.00,0.70,0.30)	5	(0.000,0.055,0.945)	0.620	0.878	620	788.95	2	898.14	5.74	14.64
	50	(0.034,0.309,0.657)		0.749		631.28	2	827.15	2.64	8.18
(0.00,0.30,0.70)	5	(0.000,0.048,0.952)	0.780	0.881	780	829.41	2	941.30	2.26	9.48
	50	(0.000,0.300,0.700)		0.780		780.00	1	1000.00	0.00	0.00

Table A.2: Simulation Results for $N = 1{,}000$, $D = 5$, $\mathbf{u}_a$ ($g = 0.05, f = 0.25$), and $\mathbf{q} = \mathbf{r} = (0.10, 0.50, 0.90)$

$\mathbf{s}_0$	T	$\overline{\mathbf{s}^*}$	$r(\mathbf{s}_0)$	$\overline{r(\mathbf{s}^*)}$	$\lfloor N \rfloor \cdot r(\mathbf{s}_0)$	$V_N(\mathbf{s}_0)$	F	$\overline{\lfloor B^* \rfloor}$	$\overline{\Delta^*}$	$\overline{\mathrm{T}^*}$
(1.00,0.00,0.00)	5	(0.000,0.055,0.945)	0.100	0.878	100	770.31	3	876.71	8.62	16.02
	50	(0.163,0.229,0.608)		0.678		482.09	3	688.58	8.83	5.35
(0.00,0.70,0.30)	5	(0.000,0.055,0.945)	0.620	0.878	620	788.95	2	898.14	5.74	14.64
	50	(0.000,0.700,0.300)		0.620		620.00	1	1000.00	0.00	0.00
(0.00,0.30,0.70)	5	(0.000,0.048,0.952)	0.780	0.881	780	829.41	2	941.30	2.26	9.48
	50	(0.000,0.300,0.700)		0.780		780.00	1	1000.00	0.00	0.00

Table A.3: Simulation Results for $N = 1{,}000$, $T = D = 5$, $\mathbf{q} = \mathbf{r} = (0.10, 0.50, 0.90)$, $\mathbf{u}_a$ ($g = 0.05, f = 0.25$), and $\mathbf{u}_b$ ($g = 0.05, f = 0.75$)

$\mathbf{s}_0$	$\mathbf{u}$	$\overline{\mathbf{s}^*}$	$r(\mathbf{s}_0)$	$\overline{r(\mathbf{s}^*)}$	$\lfloor N \rfloor \cdot r(\mathbf{s}_0)$	$V_N(\mathbf{s}_0)$	F	$\overline{\lfloor B^* \rfloor}$	$\overline{\Delta^*}$	$\overline{\mathrm{T}^*}$
(1.00,0.00,0.00)	$\mathbf{u}_a$	(0.000,0.055,0.945)	0.100	0.878	100	770.31	3	876.71	8.62	16.02
	$\mathbf{u}_b$	(0.000,0.039,0.960)		0.884		848.14	3	959.32	3.56	4.58
(0.00,0.70,0.30)	$\mathbf{u}_a$	(0.000,0.055,0.945)	0.620	0.878	620	788.95	2	898.14	5.74	14.64
	$\mathbf{u}_b$	(0.002,0.032,0.966)		0.886		860.23	3	971.27	2.14	3.61
(0.00,0.30,0.70)	$\mathbf{u}_a$	(0.000,0.048,0.952)	0.780	0.881	780	829.41	2	941.30	2.26	9.48
	$\mathbf{u}_b$	(0.000,0.028,0.972)		0.889		859.76	3	966.96	2.09	4.51

Table A.4: Simulation Results for $N = 1{,}000$, $T = D = 5$, and $\mathbf{u}_b$ ($g = 0.05, f = 0.75$)

$\mathbf{s}_0$	$\mathbf{q} = \mathbf{r}$	$\overline{\mathbf{s}^*}$	$r(\mathbf{s}_0)$	$\overline{r(\mathbf{s}^*)}$	$\lfloor N \rfloor \cdot r(\mathbf{s}_0)$	$V_N(\mathbf{s}_0)$	F	$\overline{\lfloor B^* \rfloor}$	$\overline{\Delta^*}$	$\overline{\mathrm{T}^*}$
(1.00,0.00,0.00)	(0.10,0.30,0.50)	(0.009,0.087,0.904)	0.100	0.479	100	451.56	3	942.84	4.43	7.01
	(0.80,0.85,0.90)	(0.210,0.163,0.628)	0.800	0.871	800	862.32	3	990.00	2.00	0.00
(0.00,0.70,0.30)	(0.10,0.30,0.50)	(0.011,0.060,0.929)	0.360	0.484	360	460.63	3	952.40	2.61	6.91
	(0.80,0.85,0.90)	(0.220,0.071,0.727)	0.865	0.876	865	871.94	3	995.00	1.00	0.00
(0.00,0.30,0.70)	(0.10,0.30,0.50)	(0.009,0.068,0.923)	0.440	0.483	440	460.47	3	953.80	2.55	6.69
	(0.80,0.85,0.90)	(0.000,0.300,0.700)	0.885	0.885	885	885.00	1	1000.00	0.00	0.00

Table A.5: Simulation Results for $N = 1{,}000$, $T = 5$, $\mathbf{u}_b$ ($g = 0.05, f = 0.75$), and $\mathbf{q} = \mathbf{r} = (0.80, 0.85, 0.90)$

$\mathbf{s}_0$	D	$\overline{\mathbf{s}^*}$	$r(\mathbf{s}_0)$	$\overline{r(\mathbf{s}^*)}$	$\lfloor N \rfloor \cdot r(\mathbf{s}_0)$	$V_N(\mathbf{s}_0)$	F	$\overline{\lfloor B^* \rfloor}$	$\overline{\Delta^*}$	$\overline{\mathrm{T}^*}$
(1.00,0.00,0.00)	5	(0.210,0.163,0.628)	0.800	0.871	800	862.32	3	990.00	2.00	0.00
	50	(0.287,0.356,0.356)		0.853		810.57	3	950.00	1.00	0.00
(0.00,0.70,0.30)	5	(0.220,0.071,0.727)	0.865	0.876	865	871.94	3	995.00	1.00	0.00
	50	(0.000,0.700,0.300)		0.865		865.00	1	1000.00	0.00	0.00
(0.00,0.30,0.70)	5	(0.000,0.300,0.700)	0.885	0.885	885	885.00	1	1000.00	0.00	0.00
	50	(0.000,0.300,0.700)		0.885		885.00	1	1000.00	0.00	0.00

Table A.6: Maximum Standard Errors for the Means Presented in Tables A.1-A.5

mean (table entry)	$\max(\text{standard error})$	$\max\left(\frac{\text{standard error}}{\text{mean}}\right)$
$\overline{s_1^*}$	7.7×10^{-5}	
$\overline{s_2^*}$	9.8×10^{-5}	
$\overline{s_3^*}$	6.1×10^{-5}	
$\overline{r(\mathbf{s}^*)}$	1.8×10^{-4}	2.0×10^{-4}
$\overline{\lfloor B^* \rfloor}$	2.8×10^{-2}	3.1×10^{-5}
$\overline{\Delta^*}$	1.3×10^{-2}	5.0×10^{-3}
$\overline{\mathrm{T}^*}$	2.5×10^{-2}	1.5×10^{-3}

A.9 Tables Summarizing Some Numerical Results Indicating the Effects of Acceleration of Testing

Table A.7: Results for 66 Distributions $\mathbf{s}_0$ and Parameters $N=1000$, $T=5$, $\mathbf{u}_b$ ($g=0.05, f=0.75$), $\mathbf{r}=(0.80,0.85,0.90)$, and $a=5$ (the maximum possible)

	Numbers of Cases		
	$D=5$	$D=10$	$D=50$
Unchanged Optimal Plans	10 (15.15%)	27 (40.91%)	66 (100%)
Changed Optimal Plans	56 (84.85%)	39 (59.09%)	0 (0%)
with $0<I\le5$	51	37	0
with $5<I\le10$	5	2	0

(I is the increase in the mean final number of effective systems produced by acceleration.)

Table A.8: Results for 66 Distributions $\mathbf{s}_0$ and Parameters $N = 1000$, $g = 0.05$, $T = 2.5, D = 5$, $\mathbf{r} = (0.80, 0.85, 0.90)$, and $a = 5$ (the maximum possible)

	Numbers of Cases		
	$\mathbf{u}_a$ $(f = 0.25)$	$\mathbf{u}_b$ $(f = 0.75)$	$\mathbf{u}_c$ $(f = 1.00)$
Unchanged Optimal Plans	41 (62.12%)	4 (6.06%)	62 (93.94%)
Changed Optimal Plans	25 (37.88%)	62 (93.94%)	4 (6.06%)
with $0 < I \le 5$	8	5	4
with $5 < I \le 10$	7	29	0
with $10 < I \le 15$	10	25	0
with $15 < I \le 20$	0	3	0

(I is the increase in the mean final number of effective systems produced by acceleration.)

Table A.9: Results for 66 Distributions $\mathbf{s}_0$ and Parameters $N = 1000$, $T = 2.5, D = 5$, $\mathbf{u}_b$ ($g = 0.05, f = 0.75$), and $\mathbf{r} = (0.900, 0.945, 0.990)$

	Numbers of Cases	
	$a = 7$	$a = 10$ (the maximum possible)
Unchanged Optimal Plans	5 (7.57%)	3 (4.55%)
Changed Optimal Plans	61 (92.43%)	63 (95.45%)
with $0 < I \le 5$	7	4
with $5 < I \le 10$	51	6
with $10 < I \le 15$	3	48
with $15 < I \le 20$	0	5

(I is the increase in the mean final number of effective systems produced by acceleration.)

REFERENCES

[1] M-Y Huang, D. McBeth, and S.B. Vardeman (1996), "Development Test Programs for 1-Shot System: 2-State Reliability and Binary Development-Test Results," *IEEE Transactions on Reliability*, Vol. 45, pp. 379-385.

[2] M.J. Moon, S.B. Vardeman, and D. McBeth (1999), "Development Programs for 1-Shot Systems: Decoupled Tests and Redesigns with the Possibility of Design Degradation," *IEEE Transactions on Reliability*, Vol. 48, No. 2, pp. 214-223.

[3] W.A. Nelson (1990), *Accelerated Testing: Statistical Models, Test Plans, and Data Analyses*, John Wiley & Sons, New York.

[4] W.Q. Meeker and L.A. Escobar (1998), *Statistical Methods for Reliability Data*, John Wiley & Son, New York.

[5] W.Q. Meeker and L.A. Escobar (2004), "Reliability, the Other Dimension of Quality," *Quality Technology\& Quality Management Journal*, Vol. 1, pp. - .

[6] H.A. Chan (Ed.) (2001), Accelerated Stress Testing Handbook: Guide for Achieving Quality Products, John Wiley & Sons, New York.

[7] G.K. Hobbs, (2000), Accelerated Reliability Engineering: HALT and HASS, John Wiley & Sons, New York.

[8] M.L. Cohen, J.E. Rolph, and D.L. Steffey (Ed.s) (1998), *Statistics, Testing, and Defense Acquisition: New Approaches and Methodological Improvements*, Panel on Statistical Methods for Testing and Evaluating Defense Systems, Committee on National Statistics, National Research Council, National Academies Press, Washington, D.C.

[9] D. P. Gaver and P.A. Jacobs (1997), "Testing or Fault-Finding for Reliability Growth: A Missile Destructive-Test Example," *Naval Research Logistics*, Vol. 44, pp. 623-637.

[10] J. Donovan and E. Murphy (2000), "When to Stop Development Testing: An Infrequently Used Stopping Rule Revisited," *Quality Engineering*, Vol 13, No. 3. pp. 367-376.

[11] N.S. Fard and D.L. Dietrich (1987), "A Bayes Reliability Growth Model for a Development Testing Program," *IEEE Transactions on Reliability*, Vol. 36, pp. 568-572.

[12] T.A. Mazzuchi and R. Soyer (1993), "A Bayes method for Assessing Product-Reliability During Development Testing," *IEEE Transactions on Reliability*, Vol. 43, pp. 503-510.

[13] A. Erkanli, T.A. Mazzuchi, and R. Soyer (1998), "Bayesian Computations for a Class of Reliability Growth Models," *Technometrics*, Vol. 40, pp. 14-23.

[14] S. Shevasuthisilp (2001), "Development Programs for One-Shot Systems Using Multiple State Design Reliability Models," Ph.D. Dissertation, Iowa State University.

[15] M-Y Huang (1995), "Design of Developmental Test Programs for One Shot Systems With Two State Reliability," Ph.D. Dissertation, Iowa State University.

[16] B. Senoglu and S.B. Vardeman (2004), "Development Programs for 1-Shot Systems: 2-State Reliability and Continuous (Normal) Development Test Results," *Journal of Applied Statistical Science*, Vol. 13, pp. - .