Littlewood, B. & Wright, D. (1996). Some conservative stopping rules for the operational testing of saftey-critical software (Report No. 33). Brussels: DeVa ESPRIT Long Term Research Project.

Some conservative stopping rules for the operational testing of safety-critical software

Bev Littlewood
David Wright

Centre for Software Reliability
City University
Northampton Square, London EC1V 0HB

Abstract: Operational testing, which aims to generate sequences of test cases with the same statistical properties as those that would be experienced in real operational use, can be used to obtain quantitative measures of the reliability of software. In the case of safety critical software it is common to demand that all known faults are removed. This means that if there is a failure during the operational testing, the offending fault must be identified and removed. Thus an operational test for safety critical software takes the form of a specified number of test cases (or a specified period of working) that must be executed *failure-free*. This paper addresses the problem of specifying the numbers of test cases (or time periods) required for a test, when the *previous* test has terminated as a result of a failure. It has been proposed that, after the obligatory fix of the offending fault, the software should be treated as if it were completely novel, and be required to pass exactly the same test as originally specified. The reasoning here claims to be conservative, inasmuch as no credit is given for any previous failure-free operation prior to the failure that terminated the test. We show that, in fact, this is not a conservative approach in all cases, and propose instead some new Bayesian stopping rules. We show that the degree of conservatism in stopping rules depends upon the precise way in which the reliability requirement is expressed. We show that some rules are 'completely' conservative and argue that these are also precisely the ones that should be preferred on intuitive grounds.

1 Background and motivation

The problem described here arose during recent discussions, in which one of the authors was involved, associated with the assessment of the software-based primary protection system of a nuclear reactor. The actual licensing process was based upon qualitative arguments, but the utility company volunteered to provide the regulator with a statistical demonstration that the

system's probability of failure upon demand (*pfd*) had achieved the design requirement[1] of 10^{-3}, independently of the licensing process. Here a 'demand' constitutes a set of circumstances, represented by sensor readings, that require the reactor to be shut down safely and kept thereafter in a safe state. The utility owned a simulator which it was agreed could produce input trajectories - the demands - that were statistically representative of those that the system would have to meet in real operational use. It was therefore required that the system be placed on test, be proffered 5000 demands, *and show no failures.* In the classical frequentist statistical framework, this ensures that 10^{-3} is (approximately) a 99% upper confidence bound on the true *pfd* [Parnas *et al.* 1991] (in fact 4603 perfectly executed demands gives a 99% bound); Bayesian analysis gives similar results [Miller *et al.* 1992, Littlewood & Strigini 1993].

The stopping rule here is clear: the demands will be executed until either a failure occurs, in which case the system has failed the test, or 5000 demands have been executed failure-free, in which case the system has passed the test.

The problem arises in the event that a failure *does* occur. The regulator (and, indeed, the utility) will regard the system as not acceptable, and it would be necessary to remove the bug that has been revealed before considering the system as a candidate for a further test.

Notice that, if the testing were allowed to continue (even without any attempt to remove the newly-found fault), a sufficient number of perfectly-executed demands might eventually be clocked up that even with the one failure within the total number executed we would still obtain a 99% upper confidence bound of 10^{-3} *pfd.* In fact a total of 6636 demands would be required.

The stopping rule here arises mainly for non-statistical reasons: we take the reasonable conservative position that, in view of the safety-critical nature of the system, it could not be licensed for use whilst containing a known fault, whatever the evidence that it nevertheless met the *pfd* requirement. The regulator thus insists on being shown evidence that any fault found in testing has been removed, and then argues that the only *positive* evidence in favour of the system that he/she will accept from testing is the amount of perfect working since the last fault-removal.

In our discussions of this testing scenario, the question arose: what new testing requirement should be imposed upon the system following a failure of this well-defined test? In particular, *is* it reasonable to require that the new version of the software be given exactly the same hurdle to overcome as was initially devised? The intuitive objection to such an approach is that it ignores the fact that we have received some information from the first test; in particular, in the event that the failure that terminated the test occurred after executing only a small number of demands, we have received some bad news. A regulator might reasonably be wary of ignoring such information and merely requiring the same target be achieved from the new test. The fact

---

[1] The comparative modesty of this requirement arises as a result of there being several other lines of defence against unacceptable accidents - most notably, a simple, non-software-based, secondary system. It should also be noted that this testing formed only a part of the evaluation of this software-based protection system, and did not form part of the licensing process.

that a failure has been observed early in test may, for example, be evidence of low software quality, as a result of poor production practices, or of unacceptable characteristics of the system (for example, excessive complexity). Such a wariness seems reasonable when we note that, even if discovered faults were never fixed, and regardless of the true probability of failure on demand, if we *repeatedly* put the system through tests comprising 5000 statistically representative demands, *eventually* it will succeed in passing a test.

It seems unlikely that the regulator will in reality have the luxury of demanding of the utility that it scrap the existing system and start again - if we *could* take this course, it might be reasonable to suggest that the new system should be subjected to the same acceptance criterion as the first[2]. What is needed, therefore, is a modification of the above simple stopping rule that allows failures to occur during test, and consequent removal of the faults that these reveal, but which takes into account this potential 'bad news' in specifying the further number of failure-free demands that must be executed.

The intention here is to devise stopping rules that retain the flavour of the one described above. That is, we require a stopping rule that allows the parties at any time during a test to agree that 'if the system executes *n* further demands without failure it will be deemed to have passed the test, otherwise testing must continue.'

More formally, we require a rule of the following kind:

1 At the start of test, we compute the number, n_1, of demands that must be executed failure-free for the test to succeed and stop.

2 The system is put on test and either successfully executes the n_1 demands, in which case the test stops and the system is declared to have achieved its *pfd* requirement, or a failure is observed on demand s_1 ($<n_1$), in which case the test is stopped.

3 In the light of the evidence of one failure in s_1 demands, we compute the number, n_2, of *further* demands that must be executed failure-free for the next test to succeed and stop.

4 The system is put on test again and either successfully executes the n_2 demands, in which case the test stops and the system is declared to have achieved its *pfd* requirement, or a failure is observed on demand $s_1 + s_2$ ($s_2<n_2$), in which case this test is stopped.

And so on. Clearly, it is not certain that in practice the process will terminate. This is in accord with intuition, since termination implies that the system is finally acceptable and this may never

---

[2] Although even here it might be argued that there should be some carry-over of evidence. For example, if the same development team were used, we might think that their failure to make the first system pass the test was evidence that they were not greatly competent, and this might depress our expectations of their likely success on their second try. Equally, the first failure might make us believe more strongly that the problem being tackled in building the system is a 'difficult' one, and that any putative solution is thus more likely to fail.

be the case. Essentially what is happening is a competition between the 'good news' represented by the most recent failure-free executions, and the information coming from the accumulating failures.

In the next section we present simple Bayesian solutions to this problem which retain some of the conservatism that a regulator might desire. This deals with the problem in the context of a demand-based system, as above; Section 3 treats the similar problem concerning the testing of a system that is required to operate continuously.

2 Stopping rules for demand-based systems

We shall assume that successive demands are statistically independent Bernoulli trials. Let p be the probability of failure on demand. Thus, given p, the number of failures in n demands, R, has a Binomial distribution:

$$P(R = r) = {}^{n}C_{r}\, p^{r}(1-p)^{n-r} \tag{1}$$

and in particular

$$P(R = 0) = (1-p)^{n} \tag{2}$$

Within the Bayesian framework we represent our *a priori* knowledge about the parameter of interest, here p, by the prior distribution. There are advantages in using a prior distribution from the conjugate family[3] which in this case is the Beta(a,b) distribution:

$$f(p) = \frac{p^{a-1}(1-p)^{b-1}}{B(a,b)} \tag{3}$$

where $B(a,b)$ is the Beta function and $a>0$, $b>0$ are chosen by 'you' to represent 'your' belief about p prior to seeing any test results.

In some cases it might be possible to use information about the system and its development process to give numerical values for a and b. Here we shall concentrate on the case where no such information is available, and use the 'ignorance prior' with $a=b=1$:

$$f(p) = 1 \tag{4}$$

If the system has executed n demands, and we have seen r failures, the posterior distribution of p is Beta($a+r$,$b+n-r$):

$$f(p|r,n,a,b) = \frac{p^{a+r-1}(1-p)^{b+n-r-1}}{B(a+r,b+n-r)} \tag{5}$$

---

[3] The conjugate family has the property that both prior and posterior distribution will be members of the same parametric family of distributions. It represents a kind of homogeneity in the way in which our beliefs are represented, and how they change as we receive extra information.

which reduces to

$$f(p|r,n,1,1) = \frac{p^r(1-p)^{n-r}}{B(1+r,1+n-r)} \tag{6}$$

for the ignorance prior.

2.1 A pfd-based stopping rule

We now compute n_1 by asking what is the minimum number of demands that, if executed without failure, would allow us to conclude that the system had met its *pfd* target. For the case discussed earlier, the requirement could be framed in the Bayesian context as

$$P(p < 10^{-3}) \geq 0.99 \tag{7}$$

More generally we could express the requirement as a pair (p_0, α) such that

$$P(p < p_0) \geq 1 - \alpha \tag{8}$$

From (6), n_1 is the smallest value of n for which

$$\int_0^{p_0} \frac{(1-p)^n dp}{B(1,1+n)} \geq 1 - \alpha \tag{9}$$

If the system is placed on test and failure *actually* occurs after s_1 ($<n_1$) demands, we compute n_2, the number of further demands that must *now* be executed failure-free to satisfy the reliability requirement, as follows. The posterior distribution for p immediately following the failure on the s_1 th demand is

$$f(p|1,s_1,1,1) = \frac{p(1-p)^{s_1-1}}{B(2,s_1)} \tag{10}$$

which becomes our *prior* distribution for p for the further testing that will be conducted. To compute n_2, we need the posterior distribution after seeing n_2 further demands all executed failure-free; this is

$$f(p|1,s_1+n_2,1,1) = \frac{p(1-p)^{s_1+n_2-1}}{B(2,s_1+n_2)} \tag{11}$$

Notice that this is simply the posterior distribution after seeing *both* (s_1-1) failure-free demands, followed by a failure, *and then* n_2 further failure-free demands. In fact this posterior distribution will be the same *whenever* the single failure occurred among the s_1+n_2 demands: it depends only upon the total number of demands, and the number of failures. Now n_2 is the smallest value of n for which

$$\int_0^{p_0} \frac{p(1-p)^{s_1+n-1} dp}{B(2,s_1+n)} \geq 1 - \alpha \tag{12}$$

This process continues. In general, if we have just seen the *j*th failure, and the failures occurred on the s_1th, (s_1+s_2)th, . . , $(s_1+s_2$ . . $+s_j)$th demands, we should require a further n_{j+1} demands to be executed failure-free, where n_{j+1} is the smallest value of *n* for which

$$\int_0^{p_0} \frac{p^j (1-p)^{\sum_1^j s_i + n - j}}{B(j+1, \sum_1^j s_i + n - j + 1)} dp \geq 1 - \alpha \qquad (13)$$

For the example we considered earlier, where $(p_0, \alpha) = (0.001, 0.01)$, we find from (9) that n_1=4602. Let us assume now that this first test fails, i.e. there is a failure before 4602 demands have been exercised. If this failure occurs on the 1000th demand, i.e. s_1=1000, from (12) we find that the length of the further test required after the fault has been removed is n_2=5635. In other words, a failure during the first test occurring as early as the 1000th demand suggests that we should be wary of this system and demand that the test it be required to pass after the removal of the offending fault be more stringent than the original test. The worst situation of all would be if the failure in the first test occurred on the first demand, i.e. s_1=1, in which case n_2=6634. The *best* news that it is possible to obtain from the test, short of passing it by correctly executing the 4602 demands, is for a failed demand to occur for the first time on the 4602nd demand, i.e. s_1=4602, in which case n_2=2033.

Number of failures, *j*	Total number of demands, *N*
0	4602
1	6635
2	8402
3	10041
4	11600
5	13104
6	14566
7	15995
8	17397
9	18778

Table 1: Total number of demands, *N*, needed if there have been exactly *j* failed demands, so as to claim $(p_0, \alpha) = (0.001, 0.01)$.

What happens here is that a failure *early* in the test is bad news, and the next test needs to be correspondingly longer in order that we have the required confidence in the reliability of the system. A failure occurring *late* in the test, on the other hand, does not completely outweigh the confidence that we have gained from the previous failure-free working, and so the length of failure-free working required in the next test is correspondingly reduced. The break-even point

occurs when s_1=2033, in which case n_2=4602 and the second test has the same length as the first.

Notice the symmetry of this result with that of s_1=4602 and n_2=2033 in the preceding paragraph: this results from the fact that in this model we draw the same conclusion from having seen 1 failure in 6635 (i.e. 2033+4602) demands *regardless of when this failure occurred during the sequence of demands*. This observation allows us to simplify the computations in order to use this stopping rule in practice. It is not necessary to carry out the incomplete Beta computations of (9), (12) and (13) dynamically as we successively observe particular s_1, s_2, . . Instead we need only compute, for a particular (p_0, α) reliability requirement, the total number of demands (failure-free and failed) that must be observed for j failed demands to satisfy the requirement (for j = 1, 2, . .). Table 1 shows these numbers for the case $(p_0, \alpha) = (0.001, 0.01)$ considered earlier.

This table can be computed before the test(s) are carried out. As failures occur the lengths of subsequent tests are computed as follows. Suppose the first failure occurs on the 1200th demand, i.e. s_1=1200. From the Table we see that a minimum 6635 demands need to be executed in total if one failed demand is to be allowed. Thus a further n_2=5335 demands need to be executed failure-free following the removal of the fault associated with this first failure. If now the second test ends in failure after a further 2500 demands, i.e. s_2=2500 and there have been 3700 demands executed since testing began, then the third test requires n_3=4702 (=8402-3700) demands to be executed failure-free.

The formulation of the reliability requirement in terms of a pair (p_0, α), although analysed here within the Bayesian framework, retains the flavour of a classical, frequentist, confidence bound. However, it should be noted that the interpretation of the bound is, as usual, more natural in this Bayesian form than it is classically: when we say here that $P(p < p_0) = 1 - \alpha$ (approximately, since in practice we must stop the test after an integer number of tests, and this may correspond to a confidence slightly larger than 1-α) the probability statement really does concern the *random variable p*. This contrasts with the frequentist interpretation, in which the *bound* is the subject of the probability statement: i.e. we are asserting that, of all the bounds that we *might* have computed, a proportion $(1 - \alpha)$ will exceed the true (but unknown) *p*.

The frequentist bounds give results for the stopping rule that are very close to those obtained from the Bayesian approach if, as here, we use the uniform ignorance prior distribution. In fact it can be shown (see Appendix) that the entries in tables such as that above will exceed the Bayesian ones by precisely 1. Thus for the numerical example used here, as we have already seen, the frequentist goal is 4603 failure-free demands at the outset of the test; it is N=6636 when j=1, and so on.

2.2 A reliability prediction-based stopping rule

A weakness of the (p_0, α) formulation of the reliability goal is that it does not address directly the matter of real interest: how confident are we that this system will function adequately during its life? Merely being 99% confident that the *pfd* is smaller than 0.001 is not sufficient for us to be able to say how confident we are that the system will survive, say, the number of demands that are expected in a year. In the case of a reactor protection system, and other critical systems,

it seems imperative that we have a measure of the likelihood of unacceptable behaviour during a specified period of operation.

It seems likely that in practice this formulation of the reliability requirement is taken to mean that when the test is passed we can, for all intents and purposes, treat the *pfd* as *actually being* 10^{-3}, since the 99% confidence that it is not larger than this is the same as 'almost certainty'. This would, of course, be a dangerously misleading view. There is a 1% chance that the true *pfd* takes a value in the interval (0.001,1), and we have absolutely no information from this analysis of the overall contribution to the unreliability from this component of uncertainty. We are therefore not able to draw any conclusion about the *reliability* (in particular, the probability of surviving a certain number of demands failure-free) of the system.

It is here that the Bayesian approach is superior, since it admits a formal and rigorous theory of *prediction.* We can formulate a proper *reliability* requirement as a pair (n_0, α) for which

$$P(\text{no failures in the next } n_0 \text{ demands}) \geq 1-\alpha \tag{14}$$

The *Bayesian predictive distribution* for the number of failures R_f in the next (future) n_f demands, if we have seen r failures in the past n demands, is

$$P(R_f = r_f | r,n,a,b) = \int_0^1 P(R_f = r_f | p) f(p|r,n,a,b) dp$$

$$= \int_0^1 {}^{n_f}C_{r_f} p^{r_f} (1-p)^{n_f - r_f} \frac{p^{a+r-1}(1-p)^{b+n-r-1}}{B(a+r, b+n-r)} dp \tag{15}$$

The mean and variance of this mixed distribution are

$$E(R_f) = n_f \left(\frac{a+r}{a+b+n} \right)$$

$$Var(R_f) = n_f \left(\frac{a+r}{a+b+n} \right) \left(1 - \frac{a+r}{a+b+n} \right) \left(\frac{a+b+n+n_f}{a+b+n+1} \right)$$

and it thus has a larger spread, as expected, than a corresponding Binomial distribution because of our uncertainty about the value of p.

Similarly, the distribution of the number of further failure-free demands, say X, has a greater spread than the corresponding geometric distribution; in fact

$$E(X) = \frac{b+n-r}{a+r-1} = \mu, \text{ say}$$

$$Var(X) = \mu(\mu+1) \left(\frac{a+r}{a+r-2} \right)$$

and so the coefficient of variation is greater than for the geometric.

Now, to find the total number of demands the system needs to execute, including r failures, in order to pass the test, we put $r_f = 0$ and $n_f = n_0$ in (15), and solve for N, the smallest value of n

for which the expression (15) exceeds 1-α. We shall take the uniform prior, $a=b=1$, in what follows, as before.

In order to compare this new prediction-based stopping rule with the earlier one, we shall choose the reliability requirement (n_0, α) in order that the initial requirement for the number of failure-free demands is the same, i.e. 4602. In other words, from (15), n_0 is the largest value of n for which

$$\int_0^1 (1-p)^n \frac{(1-p)^{4602}}{B(1,1+4602)} dp \geq 1-\alpha \qquad (16)$$

Clearly there are an infinite number of (n_0, α) pairs that satisfy (16). A solution that approximates most closely to the *confidence level* of the earlier example is (46,0.009895)[4]. Another solution that is of interest is (1000,0.178476), since requiring 10^3 failure-free demands with a specified probability is 'similar to' asking for a *pfd* of 10^{-3} with a specified probability. An intermediate solution is (500,0.097982). Table 2 shows the total numbers of demands that must be executed in order to pass the test with differing numbers of failures being allowed: this table is directly comparable to Table 1.

When we compare Tables 1 and 2, it is notable how in Table 2 N increases more rapidly with j than is the case in Table 1. This is in spite of the fact that the reliability goals have been chosen to be similar at the outset: i.e. the different ways of expressing the reliability requirement represented by (n_0, α) and (p_0, α) all have in common that they will be satisfied by the same number of completely failure-free demands, 4602. It suggests that the effect of failures in the case of a test for a prediction-based requirement is more serious than for a requirement based on a bound for p. Since we would argue that these prediction-based requirements are more suitable for safety-critical systems, it appears that when failures are observed a more conservative stopping rule needs to be applied.

It is notable that in each of the three examples of the prediction-based procedure, the number of failure-free demands to be processed following observation of at least one failed demand is always more stringent than the original demand. Thus in the case of a requirement (46, .009895), even if the first failure occurs on the 4602nd demand - the most optimistic case - and thus causes the first test to fail, the second test will still require 4627 (i.e. 9229-4602) failure-free demands, i.e. more than was required for the original test. This contrasts with the previous procedure for which it was only *early* failures that increased the stringency of the second test over the first. Notice that this conservatism is increased for the other two examples of (n_0, α): in fact it can be shown that whatever numerical values we assign to (n_0, α), there *will* be conservatism here so long as $n_0>1$ (see Appendix for proof), and it will increase as n_0 increases. We feel that this conservatism of the prediction-based procedure gives it an important advantage: it accords better with our informal intuition than the *pfd* bound approach, which is only partially conservative in this sense.

---

4 Here 46 is the nearest that an integer value for n_0 gives to the α=0.01 of the earlier example.

Number of failures, j	Total number of demands, N, for (n_0, α)=(46, .009895)	Total number of demands, N, for (n_0, α)=(500, .097982)	Total number of demands, N, for (n_0, α)=(1000, .178476)
0	4602	4602	4602
1	9229	9450	9681
2	13855	14298	14766
3	18481	19147	19852
4	23107	23996	24938
5	27734	28845	30024
6	32360	33694	35111
7	36986	38543	40198
8	41612	43392	45285
9	46239	48241	50372

Table 2: Total number of demands, N, needed if there have been exactly j failed demands, so as to claim (n_0, α). Notice in each case how close to linear is the increase in N with j.

2.3 A practical consideration

In the above it is assumed that the *n* demands for a particular test will be generated sequentially by independent selection from the population of all demands, with the probabilities of selection of different demands reflecting those of operational use, *and they will be executed as they are generated*. In practice, however, it may not be convenient to execute the demands in this order.

In the case of the reactor protection system testing that motivated this work, the operational profile of demands was defined in two stages. First, some basic demand scenarios were identified: SC_1, . . . SC_k. Each scenario represents a particular type of demand, such as a pipe break involving loss of coolant. Within each scenario, the individual demands were defined via parameters, such as size and location of pipe break. The probability distribution over scenarios, and the distributions over demands (parameters) within each scenario, determine the operational profile. Successive demands are then generated independently by first selecting a scenario and then selecting a demand within a scenario, using these distributions.

In this case it was convenient to generate the demands off-line before testing began, but to execute them in batches corresponding to the different scenarios. That is, the order of execution was non-random. It is easy to see that, in this case, all *n* demands must be executed before the test terminates: we cannot terminate the test at the first failure in this non-random sequence, because this may not be the first failure in the (correct) randomly ordered sequence.

However, we can still use tables such as those above to compute the stopping rules, since these depend only upon the total number of failures experienced and the number of demands

executed, and not upon *when* these failed demands occurred. Thus, for example, if the requirement is a 99% confidence that the *pfd* is better than 10^{-3} as in Section 2.1, and one failure is observed in the first test of 4602 demands, then the next test would be of length 2033 demands (6635-4602 from Table 1). If there is a further failure in this second test, then the third test would need to be of length 1767 (i.e. 8402-6635 from Table 1). Similar reasoning can be used for the reliability prediction-based stopping rule of Section 2.2.

3 Stopping rules for continuous-time systems

In this section we develop some Bayesian stopping rules for the reliability of continuously operating software, such as that in active control systems. Thus we need first to compute the time t_1 that must be executed failure-free for us to conclude that the software has met its reliability target, and so the test can be stopped. If a failure occurs before this time has elapsed, say at time $\tau_1<t_1$, we then need to compute the *further* time, t_2, of failure-free working that the new (fixed) software must achieve for us to conclude that the target reliability has been reached and stop the test. And so on.

We shall assume that failures occur in a simple Poisson process with rate λ. Thus the number of failures, R, in time t has a Poisson distribution:

$$P(R=r)=\frac{(\lambda t)^r e^{-\lambda t}}{r!} \tag{17}$$

and in particular

$$P(R=0)=e^{-\lambda t} \tag{18}$$

The conjugate family here is the Gamma. Thus if we represent our *a priori* belief about the failure rate λ by Gamma(a, b), the posterior for λ after seeing r failures during time t is Gamma($a+r$, $b+t$):

$$p(\lambda|r,t;a,b)=\frac{(b+t)^{a+r}\lambda^{a+r-1}e^{-(b+t)\lambda}}{\Gamma(a+r)} \tag{19}$$

As usual in the Bayesian framework, there is no ‘obvious’ ignorance prior. Inspecting the roles of r and t in the posterior suggests that large parameter values represent a large amount of data, and it might therefore be concluded that a small amount of data corresponds to small parameter values - and thus ignorance corresponds to $a=b=0$. Unfortunately, this results in an improper posterior when $r=0$, which is precisely the case that interests us here. Worse, the predictive distribution for the time to next failure is also improper. In what follows, therefore, we have used the improper uniform prior distribution:

$$p(\lambda)=1 \tag{20}$$

which gives the (proper) posterior:

$$p(\lambda|r,t)=\text{Gamma}(r+1,t) \tag{21}$$

which reduces to

$$p(\lambda|0,t) = te^{-\lambda t} \tag{22}$$

in the case where r=0; this is Gamma(1,t).

3.1 A rate-based stopping rule

As in the demand-based situation of Section 2, the classical statistical approach to this problem expresses the reliability target in terms of a confidence bound. Thus we might demand that the failure rate, λ, be less than 10^{-3} with 99% confidence. In the Bayesian framework we have

$$P(\lambda < 10^{-3}|r,t;a,b) = 0.99 \tag{23}$$

Once again, notice that this is interpreted as a proper probability statement about the parameter of interest, λ, unlike the interpretation of a classical bound.

More generally we can express the reliability requirement as a pair (λ_0,α) such that

$$P(\lambda < \lambda_0|r,t;a,b) = 1-\alpha \tag{24}$$

Clearly, t_1 is the value of t satisfying

$$\int_0^{\lambda_0} p(\lambda|0,t;a,b)d\lambda = 1-\alpha \tag{25}$$

which for the case of the uniform improper prior becomes

$$\int_0^{\lambda_0} te^{-\lambda t}d\lambda = 1-\alpha$$

and so

$$t_1 = -\frac{\ln\alpha}{\lambda_0} \tag{26}$$

Thus, for example when λ_0=0.001 and α=0.01 as above, t_1=4605.17.

If a failure occurs before this time has elapsed in the first test, say at time τ_1, and after the fault has been identified and fixed the program is put on test again, the time t_2 of failure-free working that is needed to achieve the reliability target is the value of t satisfying

$$\int_0^{\lambda_0} p(\lambda|1,\tau_1+t)d\lambda = 1-\alpha$$

In the case of the uniform prior this becomes, in an obvious notation

$$\int_0^{\lambda_0} Gamma(2,\tau_1+t)d\lambda = 1-\alpha \tag{27}$$

In general, if the first j tests have terminated in a failure, the duration of the (j+1)th test, t_{j+1}, is the value of t satisfying

$$\int_0^{\lambda_0} Gamma(j+1, \tau_1 + \tau_2 + \ldots \tau_j + t) d\lambda = 1 - \alpha \quad (28)$$

Notice that this is a function only of the number of failures and the total time that the software has been on test. We are thus able to simplify the calculation of the test sizes as in Section 2.1. Table 3 shows how this is done in the case of the example above where (λ_0,α)=(0.001,0.01).

Thus if the first test terminates with a failure at τ_1=2600, the amount of failure-free working required from the second test is t_2=4038.35 (6638.35-2600). If this second test terminates with a failure after τ_2=1000, the amount of failure-free working from the third test will be t_3=4805.95 (8405.95-2600-1000), and so on.

Number of failures, j	Total elapsed time on test, t
0	4605.17
1	6638.35
2	8405.95
3	10045.12
4	11604.63
5	13108.48
6	14570.62
7	15999.96
8	17402.65
9	18783.12

Table 3: Total elapsed time on test, *t*, needed if there have been exactly *j* failed demands, so as to claim (λ_0,α)=(0.001,0.01).

In the appendix we show that this Bayesian analysis, using the uniform prior for λ, gives exactly the same numerical results for tables like this as would be obtained by the frequentist approach.

As in Section 2.1 for the demand-based system, this procedure based upon a confidence for the failure rate is not 'completely conservative': the amount of further failure-free working needed to terminate successfully a test that follows a failure may be smaller than the earlier amount. Thus, if the first test fails at τ_1=4000, the system will pass the following test if it survives failure-free for t_2=2638.35 (6638.35-4000); this is less stringent than was required initially (t_1=4605.17). In the next section we develop a stopping rule where the success criterion is expressed in terms of predictive *reliability*, rather than, as here, as a rate bound. This procedure

is completely conservative: the amounts of testing in successive tests are guaranteed to be increasing.

3.2 A reliability prediction-based stopping rule

As for the demand-based system, it seems sensible here to have the possibility of specifying the reliability target in terms of a prediction about future failure-free behaviour. Thus we could formulate the requirement as a pair (t_0,α) such that

$$P(\text{no failures in next } t_0) = 1-\alpha \tag{29}$$

Now $P(\text{no failure in next } t_0 \mid j \text{ failures in } t)$

$$= \int_0^\infty e^{-\lambda t_0}\, p(\lambda \mid j,t)\, d\lambda \;=\; \int_0^\infty e^{-\lambda t_0}\, Gamma(j+1,t)\, d\lambda \;=\; \left(\frac{t}{t+t_0}\right)^{j+1} \tag{30}$$

if we use the same uniform prior as previously. Thus, in the same notation as before, t_{j+1} is the value for which

$$\left(\frac{t}{t+t_0}\right)^{j+1} = 1-\alpha, \text{ i.e.}$$

$$t = \tau_1+\tau_2+\ldots+\tau_j+t_{j+1} = t_0\,\frac{(1-\alpha)^{1/j+1}}{1-(1-\alpha)^{1/j+1}} = t_0\left[\frac{j+1}{-\log(1-\alpha)} - \frac{1}{2} + O(1/j)\right] \tag{31}$$

as $j \to \infty$.

As in section 2, we proceed to compare this approach to the one of the previous section by choosing (t_0,α) so that t_1 takes the same value as there: i.e. t_1=4605.17. We have then

$$4605.17 = t_0\left(\frac{1-\alpha}{\alpha}\right) \tag{32}$$

From the infinite number of solutions to (32), in Table 4 we show those for (46.517,0.01), (500,0.097940), (1000,0.178407). These are chosen for similar reasons to those of Table 2 in Section 2.

This table is used exactly as before. Notice, again, that the stopping rules for the values computed in the table are completely conservative, inasmuch as the amount of failure-free working that must be observed following a failure always exceeds the amount needed for the previous test: e.g. in the first column, if the first failure occurs after precisely 4605.17 time units, the amount of failure-free working in the next test for successful completion is 4628.40 (9233.57-4605.17) time units. In the appendix we prove that the stopping rules are completely conservative in this sense.

Number of failures, j	Total elapsed time, t, for (t_0, α)=(46.517,0.01)	Total elapsed time, t, for (t_0, α)=(500,0.097940)	Total elapsed time, t, for (t_0, α)=(1000,0.178407)
0	4605.17	4605.17	4605.17
1	9233.57	9453.89	9685.78
2	13861.96	14304.05	14771.85
3	18490.36	19154.56	19859.28
4	23118.76	24005.22	24947.26
5	27747.16	28855.95	30035.51
6	32375.57	33706.72	35123.91
7	37003.97	38557.52	40212.41
8	41632.37	43408.33	45300.98
9	46260.77	48259.15	50389.60

Table 4: Total elapsed time, *t*, needed if there have been exactly *j* failed demands, so as to claim (t_0, α). As in Table 2, notice in each case how close to linear is the increase in *N* with *j*.

4 Some mathematical observations

The illustrative examples on the discrete and continuous stopping rules in the previous sections were deliberately chosen to have similar numerical goals. Readers will observe that the resulting entries in Tables 1 and 3 are also numerically close, as are those in Tables 2 and 4. This is clearly no coincidence, and arises from the relationship between the (discrete) Bernoulli process and the (continuous) Poisson process. If, in section 3, we let hours be the unit of measurement, and in section 2 define a demand to be one hour's operation, then we can see the failure process as either a sequence of independent trials with probability of failure per trial p, or a Poisson process with rate λ. Here

$$p = 1 - e^{-\lambda} \tag{33}$$

approximately, as long as λ is small (so that the chance of more than one failure within the one hour 'demand' period is negligible).

We should then expect results that are the same (except for the discrete rounding) between the discrete and continuous models when these have identical prior distributions. If λ has a Gamma(a, b) prior, then from (33)

$$p \sim \frac{b^a}{\Gamma(a)}[-\log(1-p)]^{a-1}(1-p)^{b-1} \tag{34}$$

Thus the Gamma(1, b) prior for λ corresponds to the Beta(1, b) prior for p. With these priors, the results should be almost identical between the discrete and continuous models. In fact, the

uniform priors used in both section 2 and section 3 correspond to Beta(1, 1) and Gamma(1, 0+)[5] respectively. This is an example of the obvious fact that being completely indifferent between (i.e. have uniform prior belief for) any pair of values of, say, λ does not correspond to being completely indifferent between any pair of values of $p = 1 - e^{-\lambda}$ (nor, in general, of any function of λ).

Uniform prior distributions have only been used in the previous sections to illustrate our general approach. Of course, if 'you' have genuine prior knowledge, you should represent it in a proper prior distribution. Having said that, the uniform prior used in section 3 has an interesting scale-invariance property: 'ignorance' is represented in exactly the same mathematical form, regardless of units. This carries through into the predictive distribution, so that, for example, the probability of surviving failure-free for a time *kt* given that there have been no failures in time *t*, will be a function of *k* only. Indeed all questions about future behaviour given past observations can be answered without asking the questioner what time units are involved. The stopping rule used above, using the improper prior, is a particular instance of this scale-invariance property: using the Tables above we do not need to know the units of time involved. It can be shown that this scale-invariance does not hold for stopping rules based on any proper priors (clearly the uniform prior is the only one for which *prior* beliefs are scale invariant), for example the proper Gamma(*a*,*b*) prior gives a total required testing time

$$t = \tau_1 + \tau_2 + \ldots + \tau_j + t_{j+1} = t_0 \frac{(1-\alpha)^{1/(j+a)}}{1-(1-\alpha)^{1/(j+a)}} - b \tag{35}$$

A cursory inspection of Table 4 shows that *t* is close to linear in *j*. This is a consequence of the fact that

$$\frac{(1-\alpha)^{1/(j+1)}}{1-(1-\alpha)^{1/(j+1)}} = \frac{j+1}{-\log(1-\alpha)} - \frac{1}{2} + O(1/(j+1)) \tag{36}$$

with the linear function represented by the first two terms on the right giving a very good approximation for realistic (i.e. small) values of α even for small *j*. For example, the error in using this linear approximation is less than .001% for α=0.01, even at *j*=0. Thus the successive differences in the first column of Table 4 converge rapidly to 4628.4. This can be regarded as the length of failure-free operation required of the next test, when every previous test has 'only just' failed.

5 Discussion

It could be said that the procedures outlined here are somewhat pessimistic in not giving any credit for the bug fixes that have been carried out following failures. It might be argued that it is known that after each failure, when the testing is resumed, the software will be *more* reliable

---

[5] The uniform $p(\lambda)$=1 is the limiting case of Gamma(1, *b*) as $b \to 0+$

than it was immediately prior to failure, as a result of a fault being removed. In the above analysis, in contrast, the same reliability is assumed at all stages in the testing process.

We defend this pessimism on the grounds that for safety-critical systems it is necessary to be conservative in the absence of hard evidence to the contrary. Here we would have *no* evidence of the exact contribution to the overall system unreliability made by the fault that has just been removed: in particular, there is no lower bound that we can place upon this contribution, even if we could be sure that its removal had been successful. What small empirical evidence there is about the magnitudes of the contributions made by individual software faults to overall system unreliability suggests that these can vary by several orders of magnitude, and can be *very* small [Adams 1984]. The assumption made here, therefore, that there is no improvement in the reliability as a result of even a perfect fix seems the only one that is safely conservative.

On the other hand, there is a sense in which the above procedures are *not* conservative since they do not allow for the possibility that an attempt to remove a fault may not be successful. The least serious consequence could be a simple failure in the fixing process, leaving the fault (and thus the system reliability) unchanged. This does fit into the scenario described above, but is unlikely to arise in practice since simple regression testing (in addition to the operational testing being discussed here) would detect it.

More serious is the possibility that the removal attempt itself introduces a novel fault, since this has an unbounded potential to *worsen* the reliability. It is this concern that prompted the original informal approach to this problem, in which exactly the same test was required to be passed by the repaired system as was required of the original system. It was reasoned that the repaired system was a 'new' system and it was therefore safest to regard ourselves as being completely ignorant of its reliability - just as we had been of the original system.

We have shown above that this informal approach is not, in fact, conservative. Thus for a reliability requirement expressed as a (p_0, α) *pfd* bound, or as a (λ_0, α) bound for the failure rate, in the event that failure occurs early in the test, the system ought to be required to pass a *more stringent* test than the original. The intuitive reason for this is that an early failure is evidence that the system is of 'poor quality', and correspondingly greater 'good news' will be needed to overcome this 'bad news'. This is the case whether we adopt a Bayesian or classical analysis. However, these bounds are not completely conservative in the following sense: if the failure occurs sufficiently late in the test, the requirement for the next test will be *less* stringent than the current one.

However, in the case of a requirement formulated in terms of a proper prediction of reliability, (n_0, α) in the discrete case and (t_0, α) in the continuous case, we have shown that there is even greater conservatism: the number of failure-free demands that must be observed will always be greater following a failed demand *whenever* this occurs (i.e. this conservatism does not only relate to early failures).

Notwithstanding the conservatism discussed above, it has to be admitted that the underlying model used in all of the analysis of this paper is not sufficiently conservative inasmuch as it does not admit the possibility of there being an arbitrarily large increase in the probability of failure on demand as a result of a bad fix. Instead, the model effectively assumes that the 'true' *pfd*, or the 'true' failure rate, remains the same throughout, with the earlier conservatism

discussed above relating only to our *beliefs in* (in the Bayesian analysis), or our *estimation of* (in the classical context), the *pfd* or failure rate. It seems worth trying to formulate a model that plausibly represents the fault-fixing operation, so that we can take account of the possibility that a fix may make the true reliability worse than it was before the failure occurred. This does not seem to be an easy task, and will not be addressed here.

However, it is clear that such a model will be more conservative than the approaches considered above, which brings us to the question of what is the best way forward as things stand. We believe that the second, more conservative, reliability-based approach discussed here is the more appropriate for dealing with safety-critical systems. It seems to us that the user (or in the example that motivated this work, the regulator or the utility) is interested in 'how well the system will perform' - i.e. a reliability requirement based directly upon a reliability prediction - rather than a bound on the *pfd* or failure rate. The latter, particularly in the classical frequentist context, do not allow us to say anything about the former. One might ask what one could conclude practically from an assertion that the 99% upper confidence bound on the probability of failure is 10^{-3}. It would seem very rash in a critical application to act as if the true *pfd* really *were* 10^{-3} merely because 99% represents 'high' confidence.

The advantage of the Bayesian approach is that it admits a proper *predictive distribution* for the number of failures we shall see in a specified number of future demands (or specified period of working in the continuous case), and thus a proper probability that we shall see no failures in these future demands. There is no equivalent formal theory of prediction in the classical frequentist context, and one has to resort to *ad hoc* approaches such as the 'plug in' rule, where the maximum likelihood (or some other) estimator of the *pfd* is treated as the true value and substituted into the conditional formula for the reliability function. Unlike the Bayesian approach, this takes no account of our uncertainty about the estimator, and thus cannot be regarded as sufficiently conservative for safety-critical applications.

Advocates of the classical approach to statistical inference often question the dependence of the Bayesian approach upon the prior distribution. Certainly it can be difficult to elicit 'your' prior beliefs in order to express them as a probability distribution - hence our use of the 'ignorance' uniform prior in the illustrative examples here. Whilst it is clearly better to have a prior distribution that takes account of real information that 'you' may have prior to seeing the system in operational test, the numerical results here, based on the uniform prior, are still interesting precisely because they correspond so closely to the classical ones for the (p_0, α) and (λ_0, α) cases. Thus an advocate of the frequentist approach to these stopping rules cannot, we believe, have any serious grounds for questioning the use of the uniform prior in this case. If, in addition, it is accepted that the *predictions* (n_0, α) and (t_0, α) are better ways of expressing reliability requirements, it follows that the results of Tables 2 and 4 must be preferred to those of Tables 1 and 3, *and to any frequentist equivalents of the latter*.

Acknowledgement

Some of the early results in this work were reported at FTCS-25. This early work was partially supported by the PDCS2 project (EU ESPRIT Basic Research project 6362). The results for the continuous time case, and the proofs of conservatism of the recommended approaches, were obtained in the DeVa project (EU Long Term Research project 20072). We are grateful for valuable comments on this work from Peter Bishop and Lorenzo Strigini.

References

[Adams 1984] E. N. Adams, "Optimizing preventive maintenance of software products", *IBM J. of Research and Development*, 28 (1), pp.2-14, 1984.

[Johnson & Kotz 1969] N. L. Johnson and S. Kotz, *Distributions in Statistics: Discrete Distributions*, John Wiley, New York, 1969.

[Littlewood & Strigini 1993] B. Littlewood and L. Strigini, "Assessment of ultra-high dependability for software-based systems", *CACM*, 36 (11), pp.69-80, 1993.

[Miller *et al.* 1992] W. M. Miller, L. J. Morell, R. E. Noonan, S. K. Park, D. M. Nicol, B. W. Murrill and J. M. Voas, "Estimating the probability of failure when testing reveals no failures", *IEEE Trans Software Engineering*, 18 (1) 1992.

[Parnas *et al.* 1991] D. L. Parnas, G. J. K. Asmis and J. Madey, "Assessment of safety-critical software in nuclear power plants", *Nuclear Safety*, 32 (2), pp.189-98, 1991.

Appendix

Relation between Bayesian and frequentist pfd*-based stopping rules*

We shall show here that the 'classical' solution to the stopping rule based upon (p_0, α) gives a table of numbers corresponding to Table 1, but with each entry increased by 1.

The *j*th entry in the Table, in the case of the Bayesian solution, is given by the smallest value of N, say N_1, satisfying

$$\int_0^{p_0} \frac{p^j (1-p)^{N-j}}{B(j+1, N-j+1)} dp \geq 1-\alpha$$

For the classical solution it is the smallest value of N, say N_2, satisfying

$$\sum_{r=j+1}^{N} {}^{N}C_r {p_0}^r (1-p_0)^{N-r} \geq 1-\alpha$$

Now,

$$\int_0^{p_0} \frac{p^j(1-p)^{N-j}}{B(j+1,N-j+1)}dp = \sum_{r=j+1}^{N+1} {}^{N+1}C_r {p_0}^r (1-p_0)^{N-r+1}$$

is a well-known identity obtained by applying repeated integration by parts to the left hand side [Johnson & Kotz 1969]. It follows immediately that $N_2 = N_1+1$.

Relation between Bayesian and frequentist rate-based stopping rules

To show that the results of Section 3.1, using the uniform Gamma(1,0+) prior, are identical to those that would be obtained from a classical frequentist approach, we need to prove (in the notation of (19)) that

$$\int_0^{\lambda_0} p(\lambda|j,t;1,0)d\lambda = P(X_t > j|\lambda_0),$$

that is,

$$\int_0^{\lambda_0} \frac{t^{j+1}\lambda^j e^{-\lambda t}}{j!}d\lambda = \sum_{r=j+1}^{\infty} \frac{(\lambda_0 t)^r e^{-\lambda_0 t}}{r!},$$

which again follows by repeated integration by parts of the left hand side.

Proof of conservatism of stopping rule in Section 2.2

We need to prove that the total number, N, of tests required when there have been r failures observed is a convex function of r. The proof requires some properties of the function

$$h(z) = \frac{1}{z} + \frac{1}{z+1} + \ldots + \frac{1}{z+n_0-1} = \psi(z+n_0) - \psi(z)$$

where ψ is the Digamma function $\psi(z) = \frac{d}{dz}\log\Gamma(z)$:

1. h is a decreasing, positive function of $z>0$;

2. The function $h'(z)/h(z)^2$ is a negative, strictly increasing (for $n_0 \geq 2$) function of $z > 0$.

To prove this, note that the numerator of $\frac{d}{dz}\left(\frac{h'(z)}{h(z)^2}\right)$ is $h''(z)h(z)^2 - 2h'(z)^2 h(z)$ and proving that, for $z>0$, this is strictly (for $n_0 \geq 2$) positive is equivalent to applying the Cauchy-Schwarz inequality to the pair of n_0-vectors $\left\langle \frac{1}{(z+i)^{1/2}} \right\rangle_{i=0}^{n_0-1}$ and $\left\langle \frac{1}{(z+i)^{3/2}} \right\rangle_{i=0}^{n_0-1}$.

Now our stopping rule is equivalent to choosing N just large enough so that

P(No failure in $(N+1)$th to $(N+n_0)$th demands given r failures in first n demands)

$\geq 1-\alpha$, i.e.

$$1-\alpha \leq \frac{\beta(r+1, N+n_0-r+1)}{\beta(r+1, N-r+1)} = \frac{\Gamma(N-r+1+n_0)\Gamma(N+2)}{\Gamma(N-r+1)\Gamma(N+n_0+2)}.$$

If we replace r and N by continuous variables ρ and ν, with ν defined so that equality holds in

$$\frac{\Gamma(\nu-\rho+1+n_0)\Gamma(\nu+2)}{\Gamma(\nu-\rho+1)\Gamma(\nu+n_0+2)} = 1-\alpha \tag{A}$$

we can treat ν as a function of ρ and $N(r)$ is obtained as the smallest integer greater than or equal to the value of ν corresponding to $\rho=r$.

Taking logs in (A) and differentiating with respect to ρ we can characterise the function $\nu(\rho)$ as the solution to the ordinary differential equation

$$\frac{d\nu}{d\rho} = \frac{1}{1-\frac{h(\nu+2)}{h(\nu-\rho+1)}} \tag{B}$$

with the initial conditions

$$\nu(0) = n_0(\frac{1}{\alpha}-1)-1. \tag{C}$$

To show that $\nu(\rho)$ is a convex function, we will first show that

$$0 < \frac{h(\nu+2)}{h(\nu-\rho+1)} < 1 \text{ for all } \rho > 0 \tag{D}$$

It will then only remain to verify that $\frac{h(\nu+2)}{h(\nu-\rho+1)}$ is an increasing function of ρ>0, and the convexity of $\nu(\rho)$ will follow. From property 1 of the function h, the inequalities

$$0 < \nu-\rho+1 < \nu+2$$

will give us (D). Here, since ρ is positive, the RH inequality follows trivially from the left. Suppose there is some ρ>0 for which $\nu \leq \rho$-1. By taking ρ_1 to be the least such ρ>0 we obtain a contradiction since we will have from (C) $\nu(0)$-0>-1$\geq \nu(\rho_1)-\rho_1$ with $\frac{d}{d\rho}(\nu-\rho) > 0$ throughout the interval $\rho \in (0,\rho_1)$. So (D) holds. Concerning the monotonicity of $\frac{h(\nu+2)}{h(\nu-\rho+1)}$, note first that, from (B),

$$\frac{d\nu}{d\rho} = \frac{h(\nu-\rho+1)}{h(\nu-\rho+1)-h(\nu+2)}, \quad \frac{d\nu}{d\rho}-1 = \frac{h(\nu+2)}{h(\nu-\rho+1)-h(\nu+2)}.$$

So we have, after some simplification

$$\frac{d}{d\rho}\left[\frac{h(v+2)}{h(v-\rho+1)}\right]=\frac{h'(v+2)h(v-\rho+1)^2-h(v+2)^2h'(v-\rho+1)}{\left[h(v-\rho+1)-h(v+2)\right]h(v-\rho+1)^2}$$

$$=\frac{h(v+2)^2}{h(v-\rho+1)-h(v+2)}\left[\frac{h'(v+2)}{h(v+2)^2}-\frac{h'(v-\rho+1)}{h(v-\rho+1)^2}\right] \qquad \text{(E)}$$

Under the same conditions as before the term in the parentheses in (E) is positive from property 2 of *h*. Thus the expression (E) is positive. Hence $\frac{dv}{d\rho}$ is increasing as required.

Proof of conservatism of stopping rule in Section 3.2

From (31) it can be seen that we need to show that the function

$$f(j,\alpha)=\frac{(1-\alpha)^{1/j}}{1-(1-\alpha)^{1/j}}$$

is convex, for then $f(j+1,\alpha)-f(j,\alpha)>f(j,\alpha)-f(j-1,\alpha)$ as required.

The second derivative of *f* with respect to *j* is

$$\frac{\partial^2 f(j,\alpha)}{\partial j^2}=\frac{(1-\alpha)^{1/j}\log(1-\alpha)\left(2j-2(1-\alpha)^{1/j}i+\log(1-\alpha)+(1-\alpha)^{1/j}\log(1-\alpha)\right)}{\left(1-(1-\alpha)^{1/j}\right)^3 j^4}$$

If we make the substitution

$$y=-\frac{\log(1-\alpha)}{2j},$$

which is a decreasing positive function of *j*, we get

$$\frac{\partial^2 f(j,\alpha)}{\partial j^2}=\frac{16e^{2y}y^3(1-e^{2y}+y+e^{2y}y)}{(-1+e^{2y})^3\log(1-\alpha)^2}.$$

All terms of this are obviously positive, except the bracketed expression in the numerator. This is also obviously positive except for the case $0<y<1$. An expansion using Mathematica gives

$$1-e^{2y}+y+e^{2y}y=\frac{2}{3}y^3+\frac{2}{3}y^4+\frac{2}{5}y^5+\frac{8}{45}y^6+\frac{4}{63}y^7+\frac{2}{105}y^8+O(y^9)$$

and it can be shown that the coefficient of y^n in this Taylor series is $\frac{2^{n-1}(n-2)}{n!}$ for n = 3, 4, 5, . ., which completes the proof.