Modelling Correlation among Successive Inputs in Software Dependability Analyses

A. Bondavalli¹, S. Chiaradonna¹, F. Di Giandomenico² and L. Strigini²

¹ CNUCE-CNR, Via S. Maria 36, 56126 Pisa, Italy
² IEI-CNR, Via S. Maria 46, 56126 Pisa, Italy

Abstract. We consider the dependability of programs of an iterative nature. The dependability of software structures is usually analysed using models that are strongly limited in their realism by the assumptions made to obtain mathematically tractable models and by the lack of experimental data. Among the assumptions made, the independence between the outcomes of successive executions, which is often false, may lead to significant deviations of the result obtained from the real behaviour of the program under analysis. Experiments and theoretical justifications show the existence of contiguous failure regions in the program input space and that, for many applications, the inputs often follow a trajectory of contiguous points in the input space. In this work we present a model in which dependencies among input values of successive iterations are taken into account in studying the dependability of iterative software. We consider also the possibility that repeated, non fatal failures may together cause mission failure. We evaluate the effects of these different hypotheses on 1) the probability of completing a fixed-duration mission, and 2) a performability measure.

* Version 2.0
1 Introduction

The analysis of the dependability of software structures, including those explicitly designed with the aim of tolerating faults is the subject of many papers, most recently [2, 5, 7, 10, 11]. However, the realism of the models proposed and therefore their effective utility are limited by the large number of assumptions made to obtain mathematically tractable models and by the lack of experimental data. Among these assumptions, which are quite similar in all the models proposed, one which is clearly not valid in reality is the independence among successive outcomes of repeated executions of a program, that is, the failure probability is assumed to remain constant at each iteration for the entire mission duration.

The data on which most programs operate are multi-dimensional spaces with a high number of dimensions. For example, a program may read a set of 20 floating-point numbers and have another set of 30 internal variables: it therefore works (in the terminology we use) on an input space with 50 dimensions. Experiments and theoretical justifications have shown the existence of contiguous failure regions in the program input space, i.e., connected subsets of the input space such that all the individual points in them cause the program to fail. In addition, it must be observed that in many applications, such as real-time control systems, the input sequences assume the form of trajectories where two successive inputs are very close to each other. For these reasons the inputs which originate failures of the software are very rarely isolated events but more likely grouped in clusters [1, 3, 4]. In other types of programs with repeated executions, causes for correlation can be found as well: e.g. periods of peak load in time-shared computers or in communication links could lead, through unusual timing conditions, to a high probability of errors in all the executions that take place during the peak. Last, issues of imperfect recovery (state corruption) and interactions with hardware faults further complicate the problem. For all the classes of applications to which these considerations apply, analyses of software dependability performed assuming independence among successive iterations seem to lead to results excessively diverging from the real behaviour of the analysed system, [3, 4].

Another key aspect of software dependability evaluation is the model of the effects of software failures on the controlled system. A realistic model should normally consider sequences of failures: many physical systems can tolerate “benign failures” (default, presumably safe values of the control outputs from the computer), or even plain incorrect results, if isolated or in short bursts, but a sequence of even “benign” failures such that the system is effectively without feed-back control for a while will often cause actual damage (from stopping a continuous production process to letting an airplane drift out of its safe flight envelope). Predicting the distribution of bursts would be trivial with the independence assumption, but obviously optimistic: in reality, once a first failure happens, more failures are much more likely to follow than predicted by the independence assumption.

In this paper, we try to overcome one of these limitations by proposing a more realistic evaluation model in which both correlation among successive inputs of the software and sequences of consecutive failures are taken into account. We consider a program (seen as a black box), executed repeatedly for a fixed number of iterations in a mission. We analyse the impact of these new assumptions on two of the various attributes of dependability, namely the probability of surviving missions (reliability at a certain time) and performability. Starting from a typical simplistic model we show the effects obtained by changing 2 hypotheses, first each in isolation and then together. The first is the model of the effects of failures, the other is the independence or correlation among successive inputs of the software.

The structure of the paper is as follows. In Section 2, we survey previous work for modelling correlation, then we describe the class of systems we evaluate, with the assumptions that affect our models. In Section 3 the effects on dependability of the different combinations of hypotheses are described. In Section 4, the problem of identifying proper values of parameters (distributions) for the correlation among successive iterations is discussed and the models are evaluated using some of such distributions. Section 5 contains our conclusions.
2 Background and Assumptions

2.1 Literature
The problem of modelling and evaluating the effects of correlation among the outcomes of successive iterations has been addressed by [6, 12]. [6] models the behaviour of a recovery block structure [9] composed by a primary version, an alternate version and a perfect acceptance test. Assuming that the sequence of inputs evolves in a "continuous" manner, two kinds of failure events of the primary module are distinguished:

i) **point failure**: which happens when the input sequence of the primary enters a failure region,

ii) **serial failure**: a number of consecutive failures, happening with probability 1 after the occurrence of a point failure, i.e., after that the input trajectory enters a failure region.

The number of serial failures subsequent to any point failure is a random variable. Correlation among the successive failures of the alternate are not considered since at the first (point) failure of the alternate the whole scheme fails and execution stops. From these modelling assumptions a simple Markov chain with discrete time is developed allowing an analytical evaluation of the reliability (MTTF) of the recovery blocks. [12] analyses the different forms of correlation of the recovery blocks structure, including correlation among the different alternates and among alternates and the acceptance test on the same inputs. To model the correlation among successive inputs these authors make the same assumptions as [6] including the same event set and use a SRN (Stochastic Reward Nets) model to evaluate the effects of input correlation on the MTTF.

2.2 The system
We assume an application of an iterative nature, where a *mission* is composed of a constant number $n$ of iterations of the execution of the program. At each iteration, the program accepts an input and produces an output. The outcomes of an individual iteration may be: i) **success**, i.e., the delivery of a correct result, ii) a *benign failure* of the program, i.e., an output that is not correct but does not, by itself, cause the entire mission to fail, or iii) a *catastrophic failure*, i.e., an output that causes the immediate failure of the entire mission. Of course in determining if an erroneous outcome is a benign or catastrophic failure the characteristics of the controlled system must be taken into account together with those of the program. We assume here that the execution time of the program is constant, and that as soon as an iteration is over the next iteration is started. This assumption has the same practical effect as those made in [5, 11] where the execution time was described by a combination of exponential variables and a timer was used for aborting those executions that lasted too long; using the mean duration of an iteration as though it was a constant duration yielded a satisfactory approximation.

As already mentioned, we shall show the effects on the dependability of passing from a hypothesis of statistical independence among successive input values to the case in which correlation is assumed. In this context different distributions of the number of consecutive failures will be analysed. The other hypothesis that will be changed regards accounting for sequences of failures in the definition of reliability and performability. In particular, we shall model those cases in which the mission fails not only because of a catastrophic failure but also due to a sequence longer than a given number of consecutive benign failures. A brief discussion follows to clarify the main issues characterising the context we consider.

**Failure Regions**: Failure regions are "continuous" (consisting of contiguous points) and uniformly distributed in the input space. "A priori" the probability that an input belongs to a failure region is the same for each input; it is clear that some applications should be modelled assuming a different distribution since some parts of the input space may be known to be more prone to failures than others. In [3] some two-dimensional views of fault regions (blob defects) are shown for a specific program, and a number of factors affecting the shapes of the faults were identified. The shapes can be often angular, elongated and rectangular. Since there is no evidence for choosing particular shapes on a general basis our choice will be i) guided by the necessity to simplify the modelling and ii) based on the plausibility and robustness of the models.
Input Sequence: The inputs form a "trajectory": any input value is assumed to be close to the previous one. We have a so called random or deterministic walk trajectory and a step length that must be small with respect to the size of the input space (if the step length becomes comparable to the size -in each dimension- of the input space (e.g., 50%) then, as shown in [4], we obtain uniform distribution and therefore independence). In such a context many different trajectories may be considered. Examples are 1) the next input is obtained from the previous one by modifying the values on each dimension of a random small quantity, 2) (subcase of 1) a "forward-biased" trajectory: passing from one input to the next the direction may only change slightly, 3) (subcase of 2) a trajectory of points on a straight line, at a random, small distance from each other.

Consecutive Benign Failures: We shall model the effects of sequences of benign failures such that if the sequence is equal or longer than a threshold, $n_c$, $n_c > 0$, it causes the failure of the entire mission. The hypotheses we make in modelling sequences of correlated failures are:

1) a single success before the $n_c$-th failure, will bring the system in a stable state, i.e. the memory of the previous failure sequence is immediately lost;
2) the trajectory of the input sequence is "forward-biased": passing from one input to the next the direction may vary with a small angle.
3) the failure regions are convex.

The main purpose of these assumptions is to simplify the modelling without restricting too much the class of applications that can be modelled. Actually, many control applications (e.g., radar systems or navigation systems) show "forward-biased" trajectories. Assumptions 2) and 3) constrain us to trajectories that, once they have left a failure region, are unlikely to re-enter it soon. They thus allow us to consider as a constant the probability of entering a failure region since 1) the probability of re-entering the failure region just left in a small number of iterations is small, 2) after an appropriate number of iterations the probability of re-entering that region is equal to the probability of entering any other region.

2.3 Dependability indicators
The two attributes of dependability that we will consider are the probability of surviving a mission (reliability after a certain number of executions) and the performability [5, 8, 10, 11].

The reward measure used as a basis for performability is as follows: successful executions add one unit to the value of the mission; executions producing benign failures add zero; a catastrophic failure reduces the value of the whole mission to zero. The accrued value over a mission is called $M_n$, and the expected value of this measure is evaluated.

For some critical applications the main requirement is a very low probability of failure of a mission implying a requirement of minuscule probabilities of error per execution. An alternative scenario is that of comparatively non-critical applications such as somewhat complex transaction-processing or scientific applications. Here the performability figures assume more importance. Instead of considering the probability of "mission survival" separately, one can also include it in the reward model. The reward model associated to a failed mission can be zero, as in our case, or possibly, a loss exceeding the value of a typical successful mission.

3 Models
After recalling the "simplistic" analysis, we shall consider the effects of each of our two new assumption - a positive correlation between failures in successive iterations, and the possibility for repeated "benign" failures to cause a "catastrophic" failure - in isolation and then together.

3.1 "Simplistic" assumptions
In this case, a mission consists of a sequence of $n$ iterations, and the outcomes of the individual iterations (success, "benign" failure, "catastrophic" failure, with probabilities $p_s$, $p_b$, and $p_c = 1 - p_s - p_b$, respectively) are independent events.
Probability of completing a mission
The probability of completing the mission is that of a series of \( n \) executions without catastrophic failure, \((1 - p_c)^n\).

Performability
The expected value of the reward function is: 
\[
E[M_n] = n \cdot \frac{p_s}{1 - p_c} \cdot (1 - p_c)^n 
\]
which is the product of the probability of completing a mission without a catastrophic failure \((1 - p_c)^n\) and the expected number of successes in \( n \) iterations 
\[
n \cdot \frac{p_s}{1 - p_c} 
\].

3.2 Mission failure from repeated benign failures (with independence between successive iterations)
We now assume that, although the controlled system can survive an individual benign failure of the control computer, any series of \( n_c \) or more benign failures in a row will cause the mission to fail. This is a common characteristic of continuous-control systems. In certain cases, the controlled system has enough physical inertia that an individual erroneous output from the control system will not cause the controlled system to move into a prohibited state. For such systems, the probability of an isolated failure with catastrophic consequences is zero. In other cases, although some of the failures of the control system are immediately catastrophic, others (e.g., those where the control system internally detects its own failure and outputs a "safe" value to the controlled system) are not, and only if they are repeated the resulting lack of active control may cause the controlled system to drift into a dangerous state. Of course, in either case the assumption that any sequence of up to \( n_c - 1 \) failures will be tolerated, and all longer sequences will be catastrophic, is still a simplification of reality, yet more realistic than assuming that a controlled system can tolerate any arbitrary series of "benign" failures.

Probability of completing a mission
Given the independence between successive iterations, we model each iteration of the program as having three possible outcomes: immediate catastrophic failure, with probability \( p_c \), benign failure, with probability \( p_b \), and success with probability \((1 - p_c - p_b)\). The assumption that \( n_c \) or more benign failures in a row cause mission failure of course decreases (all model parameters being equal) the probability of surviving a mission. However, a mission failure due to serial benign failures requires a sequence of at least \( n_c \) benign failures, with a probability \( p_b^{n_c} \). Hence, the probability of a mission failure due to a series of benign failures, \( p_{c\text{-serial}} \), is less than \( p_b^{n_c} \). This factor will be negligible if \( p_c \) and \( p_b \) are of the same order of magnitude, while must be taken into account if \( p_c \) is very small (e.g., because the controlled system can tolerate all erroneous control signals provided they have a short duration).

A tighter test of whether \( p_{c\text{-serial}} \) can be neglected can be obtained as follows. An upper bound on \( p_{c\text{-serial}} \) is the probability that a series of iterations without catastrophic failure is followed by a success and then \( n_c \) benign failures:
\[
\sum_{i=0}^{n-n_c-1} (1 - p_c)^i \cdot p_s \cdot p_b^{n_c} = p_s \cdot p_b^{n_c} \cdot \frac{1 - (1 - p_c)^{n-n_c}}{p_c}
\]
The total probability of mission failure is larger than \((1 - (1 - p_c))^n\), the probability of mission failure if series of benign failures are of no concern. If the upper bound on \( p_{c\text{-serial}} \) is negligible in comparison with this lower bound on the probability of mission failure, then it is legitimate to neglect \( p_{c\text{-serial}} \) in computing the latter.
Performability
With our hypothesised reward function, the only effect of the increased probability of mission failure is that a smaller proportion of the missions will be completed. Thus, the expected performability will be decreased by the same amount as the probability of completing a mission.

3.3 Correlation between successive iterations (without mission failure from repeated benign failures)
We now assume that a failure at an iteration of the program makes it more likely than otherwise that the program will fail at the next iteration as well. The system can then be modelled, for instance, by the three-state Markov chain in Figure 1, which would degenerate to the case of independence if \( p_{BB} = p_{SB}, \) \( p_{BS} = p_{SS} \). If we assume \( p_{BB} > p_{SB} \), we would expect the behaviour of the system to be worse than under the independence assumption. This worsening would be due to the fact that the marginal probability of being in the "benign failure" state has increased.

![Figure 1. The model for the iterative execution of a system with correlation.](image)

Probability of surviving the mission
There are two possible mechanisms through which this increased probability of spending time in the "benign failure" state can affect the completion of missions. We have decided not to consider - yet - the fact that a series of consecutive benign failures can cause a mission failure. The other possible mechanism would be one in which the probability of the next iteration producing a catastrophic failure increases if the last iteration produced a failure, albeit benign. This is modelled by setting \( p_{BC} > p_{SC} \) in the model. This looks like a realistic assumption in many cases: for instance, one may assume that a benign failure implies that the program has entered a region of its input space where failure in general is especially likely (as implied by the assumption of positive correlation), and that a fixed proportion of such failures happens to be immediately catastrophic. However, there are other realistic scenarios: for instance, there may be a controlled system where most erroneous control signals are immediately "catastrophic", but the control system is engineered to detect its own internal errors and then issue a safe output and reset itself to a known state from which the program is likely to proceed correctly. Then, one may well assume that most benign failures are due to this mechanism, and are very likely to be followed by a success: \( p_{BC} < p_{SC} \) and, indeed, \( p_{BB} < p_{BS} \). We will not consider this scenario any further.

It is worth repeating that these are the only two mechanisms through which positive correlation between successive failures may affect the probability of completing a mission. To analyse their separate contributions, we can consider one specific form of positive correlation between successive (benign) failures, i.e., we assume that when input trajectories cross failure regions the "length of stay" (number of execution steps before leaving the region) has a geometric distribution. This is modelled by the Markov chain in Figure 1. Then, assuming that the
random variables $X_i$ represent the state of the system at time steps $i=0,1,2,...$, and the initial state of the system is $S$, i.e. $P(X_0 = S) = 1$, we can write:

$$P(\text{mission success}) = \prod_{i=1}^{n} P(X_i \neq C \mid X_{i-1} \neq C).$$

Each term in the product has the form:

$$P(X_i \neq C \mid X_{i-1} \neq C) =
= (1 - p_{SC}) \cdot P(X_i = S \mid X_{i-1} \neq C) +
= (1 - p_{BC}) \cdot P(X_i = B \mid X_{i-1} \neq C) =
= (1 - p_{SC}) + (p_{SC} - p_{BC}) \cdot \frac{P(X_{i-1} = B)}{P(X_{i-1} \neq C)}.$$

Since the last term in the right-hand product is positive, the sign of the difference $(p_{SC} - p_{BC})$ determines whether the probability of surviving the mission is greater or lower than it would be in case of independence between successive iterations, setting $p_c = p_{SC}$, and $p_b = p_{SB}$.

**Performability**

The expected reward will be affected by the increased number of benign failures and, if $p_{BC} > p_{SC}$, also by the increased probability of not completing a mission.

### 3.4 Correlation between successive iterations, allowing mission failures from repeated benign failures

#### 3.4.1 Presentation of the model

A model of this more complex case is that in Figure 2. To model the correlation between successive failures, we assume that after an iteration with success (state $S$), the program has a probability $p_{sb}$ that the next execution will produce a benign failure (i.e., that the input trajectory has entered a failure region). However, once in a failure region, the probabilities of staying there for one, two or more iterations are given by the parameters $p_1, p_2, \ldots$. The parameters $p_{nn}$ designates the probability of staying for $n_c$ iterations or more $p_{nn} = 1 - \sum_{i=1}^{n_c-1} p_i$.

So, for instance, with probability $p_{sb}p_2$ the program enters a failure region *and* will have a series of two benign failures: in the model, it goes into state $B_2$, from which, unless a catastrophic failure occurs (arc from $B_2$ to $C$ labelled $p_{bc}$) it will be compelled to move to $B_1$, after which it exits the failure region. This explains the role of the states $B_1, B_2, \ldots$, all designating a "benign" failure of the last execution. If the sequence of benign failures were longer than $n_c-1$, a mission failure would occur. This is described by adding the term $p_{sb}p_{nn}$ to the probability of the transition from $S$ to $C$. By choosing the values of the parameters $p_i$, we assign a distribution function to the variable "number of consecutive benign failures", conditional on being inside a failure region, and on not incurring a catastrophic failure (which, from any of the states $B_1$, has probability $p_{bc}$). Notice that once in $B_1$ (the last benign failure in the crossing of a failure region), the program may move to another success, or it may enter another failure region: this is modelled by the series of downward arcs issuing from $B_1$, labelled $p_{sb}p_1, p_{sb}p_2, p_{sb}p_3, \ldots$. The probabilities on these arcs are the same as those on the downward arcs issuing from $S$ on the left: the probabilities of the trajectory entering a new failure region is independent of how long ago it left another failure region.

We point out that state $C$ models the failure of a mission due either to crossing a failure region and staying there for at least $n_c$ iterations, or to a "catastrophic" failure. A third cause for mission failure exists, i.e., crossing two or more failure regions without interruption, staying
in each for less than \( n_c \) failures but so that the total number of consecutive benign failures exceeds \( n_c - 1 \). This mechanism is not represented by a state in our chain, but rather by a trajectory which, after entering one of the \( B_i \) states, and stepping up all the way to \( B_1 \), takes one of the downward arcs on from \( B_1 \) back to one of the \( B_i \), and does so one or more times until it has spent \( n_c \) iterations in the set of the \( B_i \) states. In solving the models, we have adopted the simplification of neglecting this kind of events, and developed conditions for this simplification to be acceptable. This analysis is detailed in paragraph 3.4.2 and 4.3.

**Figure 2. The model for iterative executions with failure clustering.**

**Probability of completing a mission**

Under these conditions, the positive correlation between successive iterations will affect the probability of completing a mission both via the probability of having \( n_c \) consecutive "benign" failures and via the longer stays in failure regions, where the risk of "catastrophic" failure is greater. We can use again the expression:

\[
P(\text{mission success}) = \prod_{i=1}^{n_c} p(\text{state}_i \neq C \mid \text{state}_{i-1} \neq C)
\]

Where each term in the product now has the form:

\[
p(\text{state}_i \neq C \mid \text{state}_{i-1} \neq C) = (1 - p_{sc} - p_{nn}p_{sb}) + (p_{sc} + p_{nn}p_{sb} - p_{bc})
\]

\[
* p(\text{state}_{i-1} = B_2 \text{ or...or state}_{i-1} = B_{n_c-1} | \text{state}_{i-1} \neq C)
\]
It can be observed that the first term in this sum represents the case that sequences of benign failures do not affect the probability of mission failures. The second term represents the contribution of the higher (or lower, as the case might be) probability of catastrophic failure after a benign failure, compared to that after a successful iteration. The third term represents the probability of mission failure due to sequences of $n_c$ or more benign failures.

**Performability**

In this last case we describe the function of the performability measure that can be obtained as follows. From the definition of $M_n$ we get:

$E[M_n] = E[\text{number of successes}|\text{mission success}] \cdot P(\text{mission success})$, where 

$E[\text{number of successes}|\text{mission success}] = "the mean number of successful executions of the software given that the mission is successful"$ which can be written as:

$n - E[\#\text{crossed regions}|\text{mission success}] \cdot E[\#\text{benign failures for region}|\text{mission success}]$

where $n$ is the number of iterations in a mission, $E[\#\text{crossed regions}|\text{mission success}]$ is the mean number of failure regions visited during the mission, given that the mission is successful, and $E[\#\text{benign failures for region}|\text{mission success}]$ is the mean number of benign failures in a region, given that the mission is successful.

$E[\#\text{benign failures for region}|\text{mission success}]$ can be written as:

$$\sum_i i \cdot P(#\text{benign failures for region} = i | \text{mission success}) =$$

$$\left(\sum_{i=1}^{n-1} i \cdot p_i \cdot p_{bb}^{i-1}\right) \cdot \left(\sum_{j=1}^{n-1} p_j \cdot p_{bb}^{j-1}\right).$$

Let $NFAIL_k$, with $k=1,...,n$, be the random variable representing the number of consecutive serial failures observed after the program enters the $k$-th failure region; these variables are independent and with the same distribution "$p$", that is $P(NFAIL_k = j) = p_j$, $\forall j = 1,2,3,...$.$

$E[\#\text{crossed regions}|\text{mission success}]$ can be written as:

$$\sum_{i=0}^{n} i \cdot P("#\text{crossed regions} = i" \land "\text{mission success")}/P(\text{mission success})$$

where $P("#\text{crossed regions} = i" \land "\text{mission success")}$, in its turn, can be written as:

$$\sum_{1 \leq j_1,...,j_i \leq n_c-1} P("#\text{crossed regions} = i" \land "\text{mission success");" k=1 "NFAIL_k = j_k") \ast$$

$$P_{\bigcap_{k=1}^{i} "NFAIL_k = j_k"}.$$
Putting the pieces together it comes out that:

\[
E[M_n] = P(\text{mission success}) E[\text{number of successes}|\text{mission success}] =
= P(\text{mission success}) (n - E[\text{# crossed regions}|\text{mission success}]) 
* E[\text{# benign failures for region}|\text{mission success}]) = 
= P(\text{mission success}) \left( n - \sum_{i=0}^{n} i \frac{P("\text{# crossed regions} = i\wedge\text{mission success}")}{P(\text{mission success})} \right) 
* E[\text{# benign failures for region}|\text{mission success}]) = 
= P(\text{mission success}) \left( n - \sum_{i=0}^{n} i \frac{1}{P(\text{mission success})} \sum_{i=0}^{n} \sum_{0 \leq h \leq j \leq n+h} \left( n - (j - i - h) \right) p_{sb} (1 - p_{bc})^{j-i-h} \right) 
= P(\text{mission success}) \sum_{i=0}^{n} \sum_{0 \leq h \leq j \leq n+h} \left( n - (j - i - h) \right) p_{sb} (1 - p_{bc})^{j-i-h} 
= P(\text{mission success}) \left( \frac{1 - P_{bc}}{1 - P_{sc} - P_{sb}} \right)^{j-i-h} (1 - P_{sc} - P_{sb})^{n-i} \prod_{k=1}^{i} p_{jk} \left( \sum_{h=1}^{n_{c}-1} h p_{hb} p_{bb}^{-1} \right) \left( \sum_{h=1}^{n_{c}-1} p_{hb} p_{bb}^{-1} \right) = 
\]

To make an analysis of the expected performability in this last setting, we can observe that the expected reward will be affected by:

1) the increased number of benign failures,
2) if \( p_{bc} > p_{sc} \), also by the increased probability of not completing a mission due to a catastrophic failure, and
3) the probability of not completing a mission due to sequences of \( n_{c} \) or more benign failures.

### 3.4.2 Simplifications adopted in the model

This subsection is devoted at analysing the simplifications of the model just proposed and stating under which conditions their effects are negligible. We shall limit this analysis to the probability of mission failure (actually the performability measure is a function of this probability).

Our model does not take into account failures of the mission due to this event (simplification 1): the crossing of two or more failure regions without interruption, where the stay in each is less than \( n_{c} \) iterations but so that the total number of consecutive benign failures exceeds \( n_{c}-1 \). Let \( p_{simpl1} \) be the probability of such an event and let \( p_{ev\sup1} \) be an upper bound of it, that is \( p_{simpl1} \leq p_{ev\sup1} \).

Another simplification (simplification 2 in the following) regards the modelling of the program entering a failure region and having a series of at least \( n_{c} \) benign failures. The exact modelling of such events requires a sequence of \( n_{c} \) states (including \( C \)), one for each iteration. The purpose would be to distinguish if in an iteration the failure is a benign one, in which case the program would continue in the sequence leading eventually to \( C \), or if that is a catastrophic
failure in which case no more iterations would be performed and the C state is immediately reached. Obviously if the mission terminates while in one of these states, excluding state C, (i.e., the n-th iteration of the mission leads to one of these states) the mission is successful. In our model we consider just one transition from \( S \) to C (and from \( B_1 \) to C), labelled \( p_{sb}p_{nn} \). Therefore there are in the real system some missions which terminate successfully (because the last iteration brings the system to one of these intermediate states) that are modelled as failed. Let \( p({\text{simp1/2}}) \) be the probability of mission failure due to the simplification 2 and let \( p_{evsup2} \) an upper bound of it, that is \( p({\text{simp1/2}}) \leq p_{evsup2} \).

The two simplifications have opposite effects: while simplification 1 does not consider some of the possible failures of the system, simplification 2 considers as failures some events that are not failures. Denoting with \( P(\text{mission failure}) \) the probability of failure of the mission derived from the model, it is possible to bound the real failure probability in the interval:
\[
(P(\text{mission failure}) - p_{evsup2}, P(\text{mission failure}) + p_{evsup1}].
\]

In the following we shall derive the expressions for \( p_{evsup1} \) and \( p_{evsup2} \) and in the next section we will try to identify the conditions allowing to consider \( \max\{p_{evsup1}, p_{evsup2}\} \) negligible with respect to \( P(\text{mission failure}) \).

An obvious upper bound for \( p({\text{simp1}}) \) is given by the probability of the event: "the program crosses at least two failure regions without interruption", which in its turn is less than the following expression (as can be observed from Figure 1):
\[
\sum_{i=0}^{n-1} P(\text{state}_i = S) \times p_{sb} \sum_{j=1}^{n_c-1} p_j \times p_{bb} \times p_{sb}, \quad \text{where } P(\text{state}_i = S) \text{ is the probability that "the program enters state } S \text{ at the } i\text{-th iteration, with } i=0,1,...,n-1" \text{ and } p_{sb} \sum_{j=1}^{n_c-1} p_j \times p_{bb} \times p_{sb} \text{ is the conditioned probability that "starting from state } S \text{ the program crosses at least two failure regions without interruption".}
\]

Using again the expressions \( P(\text{state}_i \neq C) = \prod_{j=1}^{i} P(\text{state}_j \neq C \mid \text{state}_{j-1} \neq C) \) derived in the paragraph 3.4.1, we obtain:
\[
P(\text{state}_i = S) \leq P(\text{state}_i = S \text{ or state}_i = B_1 \text{ or...or state}_i = B_{n_c-1}) \\
\quad \leq P(\text{state}_i \neq C) \\
\quad \leq (1 - p_{sc} + \max\{0, p_{sc} - p_{bc}\})^i.
\]

From which:
\[
p(\text{simp1}) \leq p_{evsup1} = \\
\quad = \sum_{i=0}^{n-1} (1 - p_{sc} + \max\{0, p_{sc} - p_{bc}\})^i \times p_{sb} \sum_{j=1}^{n_c-1} p_j \times p_{bb} \times p_{sb} \\
\quad = p_{sb} \sum_{j=1}^{n_c-1} p_j \times p_{bb} \times p_{sb} \sum_{i=0}^{n-1} (1 - p_{sc} + \max\{0, p_{sc} - p_{bc}\})^i \\
\quad = (1 - (1 - p_{sc} + \max\{0, p_{sc} - p_{bc}\})^n) \times \frac{p_{sb}}{p_{sc} - \max\{0, p_{sc} - p_{bc}\}} \sum_{j=1}^{n_c-1} p_j \times p_{bb} \times p_{sb} \times (1 - p_{sc} + \max\{0, p_{sc} - p_{bc}\})^i.
\]
Now we derive some expression for \( p_{evsup2} \). We start from the expression for \( p(simp12) \) which is given by:

\[
\sum_{i=0}^{n_c-2} P(\text{state}_n-i-1 = S' \text{ or } \text{state}_n-i-1 = B_1')*p_{sb}*p_{nn}*p_{bb}^i,
\]

with \( P(\text{state}_n-i-1 = S' \text{ or } \text{state}_n-i-1 = B_1')*p_{sb}*p_{nn} \) representing the probability of the event 'the program enters a failure region at the iteration 'n-i' (i.e., only 'i' iterations remain to be performed before the end of the mission), and will stay in it for at least \( n_c \) iterations, with \( i=0,1,...,n_c-2 \) and \( p_{bb}^i \) representing the probability that the first 'i' failures of this sequence are all benign. Following an approach similar that used in deriving \( p_{evsup1} \), we obtain:

\[
P(\text{state}_n-i-1 = S' \text{ or } \text{state}_n-i-1 = B_1') \leq (1-p_{sc}+\max\{0,p_{sc}-p_{bc}\})^{n-i-1}.
\]

From which:

\[
p(simp12) \leq p_{evsup2} = \sum_{i=0}^{n_c-2} (1-p_{sc}+\max\{0,p_{sc}-p_{bc}\})^{n-i-1}*p_{sb}*p_{nn}*p_{bb}^i
\]

\[
= p_{sb}*p_{nn}*(1-p_{sc}+\max\{0,p_{sc}-p_{bc}\})^{n-1}*\sum_{i=0}^{n_c-2} \left( \frac{p_{bb}}{1-p_{sc}+\max\{0,p_{sc}-p_{bc}\}} \right)^i.
\]

In the particular setting where \( p_{sc} = p_{bc} \), \( p_{evsup2} \) reduces to: \( p_{sb}*p_{nn}*(1-p_{sc})^{n-1}*(n_c-1) \); otherwise we obtain:

\[
p_{evsup2} = p_{sb}*p_{nn}*(1-p_{sc}+\max\{0,p_{sc}-p_{bc}\})^{n-1} * \left( \frac{p_{bb}}{1-p_{sc}+\max\{0,p_{sc}-p_{bc}\}} \right)^{n_c-1} + \left( \frac{1-p_{bc}}{1-p_{sc}+\max\{0,p_{sc}-p_{bc}\}} \right)^{n_c-1}.
\]

4. Evaluations results

The model proposed in Section 3.4 is a general one in the sense that it is not tied to any specific distribution of the length of stays in failure regions. As there is no evidence for choosing a particular distribution on a general basis, we show the effect of a few different distribution functions, before discussing the properties that they share.

4.1 Distribution functions for the length of stay in a failure region.

The distribution functions we consider are: the geometric, a modified negative binomial (which includes the geometric as a particular case), a modified Poisson distribution and an ad hoc distribution described in the following.

4.1.1 Modified geometric, modified negative binomial and modified Poisson

The geometric distribution, defined as \( p_i = q*(1-q)^i \), for some \( q \in (0,1) \), fits very well contexts where most failure regions are of small size. It seems suitable in context
where: a) high-quality software is used, meaning that the residual failure regions are of small size and so the input trajectory will remain in the failure region for just a few iterations; b) large failure regions can still be present, but the probability that the input trajectory will enter them and stay for many iterations is negligible. Regions of this kind could be found, for example, when the failure depends only on a few values of one dimension of a multidimensional input space. The geometric distribution is memoryless. It models trajectories having at each iteration the same probability \( q \) of leaving the failure region, independently of how long they had been in it before.

If the probability of entering a large failure region and staying in it for a considerable number of iterations is not negligible compared to the probability of entering small failure regions, a modified negative binomial distribution function and a modified Poisson distribution function seem to be more appropriate.

The modified negative binomial is defined as
\[
p_i = \binom{i + r - 2}{r - 1} q^r (1 - q)^{i-1}, \quad i = 1, 2, 3, \ldots
\]
for some \( r = 1, 2, 3, \ldots \) and \( q \in (0, 1) \). The modified Poisson is defined as
\[
p_i = \frac{e^{-\alpha} \alpha^{i-1}}{(i-1)!}, \quad i = 1, 2, 3, \ldots
\]
for some \( \alpha > 0 \). In the evaluation performed, the modified negative binomial is used with parameter \( r = 5 \).

### 4.1.2 An ad-hoc distribution.
A further distribution is considered as an example on how ad hoc distribution functions can be derived based on knowledge available in particular cases. Suppose that for a particular application the following knowledge has been obtained:

1. the input space is a discrete two-dimensional (Cartesian) space;
2. the shape of failure regions is that of a square with its sides parallel to the axes of the input space;
3. the input trajectory is a directed straight line crossing the square region failures vertically, horizontally or diagonally.

This ad-hoc distribution can therefore be defined as:
\[
p_i = \frac{i + 1}{3i - 1} p_L(i) + \sum_{j=i+1}^{\text{max} L} \frac{2}{3j - 1} p_L(j),
\]
with \( i = 1, 2, 3, \ldots \), where: \( p_L(j) \) is the probability that the input trajectory enters a failure region having the side length equal to \( j \), \( 1 \leq j \leq \text{max} L \), and \( \text{max} L \) is the maximum length of the side of a failure region. The expressions \( \frac{i + 1}{3i - 1} \) and \( \frac{2}{3j - 1} \) are the probabilities that the length of stays in failure regions be \( i \), conditional on being inside a failure region having the side length equal to respectively \( i \) and \( j \).

Different distributions could be considered for \( p_L \); among these the truncated geometric represents an input space mainly populated by failure regions of small size. In the following numerical evaluation, the length of the sides of the failure regions ranges between 1 and 30.

### 4.2 Evaluation and discussion
Now we show the results for the probability of mission failure and the performability measure obtained from the model in which correlation between successive iterations, and mission failures from repeated benign failures are taken into account. We use the four distributions described previously to model the correlation among successive inputs and a set of plausible values for the model parameters, as shown in Table 1. The number of iterations in a mission \( n \), is equal to \( 10^6 \) (a realistic number, e.g., for civil avionics where the average duration of one iteration could be 20-50 milliseconds and the duration of the whole mission could be around 10 hours).
Parameters and their values
\[ p_{sb} = 10^{-5} \]
\[ p_{sc} = 10^{-9} \]
\[ p_{ss} = 1 - p_{sb} - p_{sc} \]
\[ p_{bc} = 10^{-3} \]
\[ p_{bb} = 1 - p_{bc} \]
\[ n_c = 10 \]

(Table 1. Parameter values used in numerical evaluation.)

The two factors that presumably have the greatest influence on the probability of mission failure (that is, the probability of entering state C in Figure 2) and on the performability are 1) the probability of exceeding a sequence of \( n_c - 1 \) consecutive benign failure, \( p_{nn} \) and 2) the mean stay in a failure region, once the input trajectory enters it. We shall therefore evaluate the variations in the dependability figures as a function of these two factors, while keeping all others constant.

\[ P[C] \]
\[ E[M_n](x 10^3) \]

(Figure 3. Probability of mission failure (a) and performability measure (b) as a function of \( p_{nn} \).)

In Figure 3a and 3b, showing, respectively, the behaviour of the probability of failure and the performability as functions of the probability of exceeding \( n_c - 1 \), two additional distribution functions \( p^* \) and \( p^{**} \) have been introduced. Once a value for \( p_{nn} \) has been fixed, \( p^{**} \), defined such that \( \sum_{i=1}^{n_c-2} p^{**}(i) = 0 \), \( p^{**}(n_c-1) = 1 - p_{nn} \), and \( \sum_{i>n_c-1} p^{**}(i) = p_{nn} \), represents the worst case behaviour of an input trajectory: the case in which the input trajectory, once in a failure region, stays in it for at least \( n_c - 1 \) iterations; while \( p^* \), defined such that \( p^*(1) = 1 - p_{nn} \), \( \sum_{i=2}^{n_c-1} p^*(i) = 0 \), and \( \sum_{i>n_c-1} p^*(i) = p_{nn} \), represents the best case behaviour in which, once in a failure region, the trajectory may either exit immediately (after one benign failure) or stay in it for at least \( n_c \) iterations. The range of \( p_{nn} \) has been limited between 0 and \( 10^{-2} \) because higher values would imply a probability of mission failure too high for being acceptable.

A few observations can be derived related to Figure 3:

1) \( p^* \) shows better figures than \( p^{**} \) because \( p_{bc} \) was chosen to be higher than \( p_{sc} \), in the other case the opposite would have been true. Moreover, increasing \( n_c \) increases the difference between the probabilities of mission failure implied by the two extreme distributions \( p^* \) and \( p^{**} \);

2) the distance of curves given by the considered distributions from the curve given by \( p^* \) depends on the mean stay in the failure regions (and on the difference \( p_{bc} - p_{sc} \)).
the value of $p_{sb}$ determines the slope of the curves; higher values imply that the probability of entering a failure region becomes higher and therefore the probability of mission failure increases;

4) as expected, the four distributions we considered are all included in between the two extreme distribution functions and are closer to $p^*$, since they have been chosen to model a higher probability of short sequences (than of long ones) of serial failures;

5) the definition of $p^*$ and $p^{**}$ allows simple tests on the viability of specific applications requiring only an estimate of $p_{mn}$. The designer of a software application can bound the probability of mission failure and the performability, using $p^*$ and $p^{**}$. If the worse of the two values obtained is sufficient to satisfy the application requirements, further information regarding the actual distribution becomes unnecessary.

Figure 4a and 4b show, respectively, the behaviour of the probability of failure and the performability as function of the mean stay in a failure region.

Figure 4. Probability of mission failure (a) and performability measure (b) as a function of the mean stay in a failure region.

Analysing them we observed that, with the same mean, the distributions with higher variance cause worse behaviour. The figures obtained from the ad hoc distribution are very similar to those obtained from the other distributions. The plots have been made for a range of parameters that extends to unrealistic situations: with the set of parameter values chosen, our plots show that, to obtain probabilities of mission failure up to $10^{-1}$, the mean stay in failure regions must be limited to 2-3.

4.3 Evaluation of the error introduced with the simplifications

We now evaluate, for the selected values of the parameters, the ratio between the upper bounds $p_{ev, sup 1}$ and $p_{ev, sup 2}$ and $P(\text{mission failure})$.

In our numerical analysis $p_{bc} > p_{sc}$; this allows to simplify $p_{ev, sup 1}$ with

$$\left(1-(1-p_{sc})^n\right) \frac{p_{sb}^2 n c^{-1}}{p_{sc}} \sum_{j=1}^{n_c-1} p_j p_{bb}^j = 1-(1-p_{sc})^n$$

where $1-(1-p_{sc})^n$ represents the probability of mission failure in the case that the outcomes of the individual iterations are independent events (as showed in section 3.1, assuming probabilities $p_c = p_{SC}$, and $p_b = p_{SB}$).

In 3.4.1 we derived that $P(\text{mission failure}) \geq 1-(1-p_{sc})^n = 0.0009995$. So the ratio between $p_{ev, sup 1}$ and $P(\text{mission failure})$ is less than or equal to

15
\[
\frac{P_{evsup1}}{(1-(1-p_{sc})^n)} = \frac{P_{sb}^2}{p_{sc}} \sum_{j=1}^{n_{c}-1} p_j * p_{bb}^{j-1}. \text{ Observing that } p_{bb}^{j-1} \leq 1 \text{ and } \sum_{j=1}^{n_{c}-1} p_j = 1 - p_{nn} \text{ we obtain:}
\]
\[
\frac{P_{sb}^2}{p_{sc}} \sum_{j=1}^{n_{c}-1} p_j * p_{bb}^{j-1} \leq \frac{P_{sb}^2}{p_{sc}} \sum_{j=1}^{n_{c}-1} p_j = \frac{P_{sb}^2}{p_{sc}} (1 - p_{nn}) \leq \frac{P_{sb}^2}{p_{sc}} = 0.1. \text{ This upper bound of the error introduced by simplification 1, is valid for any distribution of the serial failures. The main factors affecting this error are } p_{sb} \text{ and } p_{sc} \text{ as it can be observed looking at the final expression. A tighter upper bound is shown in Figure 5, that plots the ratio}
\]
\[
\left( (1-(1-p_{sc})^n) \frac{P_{sb}^2}{p_{sc}} \right) / P(\text{mission failure}) \text{ for the range } [1-(1-p_{sc})^n, 0.1] \text{ of } P(\text{mission failure}). \text{ An even better upper bound could be obtained by evaluating the values taken by } p_{evsup1} \text{ according to the various distributions.}
\]

![Figure 5. Upper bound of the ratio between $p_{evsup1}$ and $P(\text{mission failure})$.](image)

Still considering that $P(\text{mission failure}) \geq 1-(1-p_{sc})^n$, the ratio between $p_{evsup2}$ and $P(\text{mission failure})$ is less than or equal to
\[
\frac{p_{sb} * p_{nn} * (1-p_{sc})^n * \left( 1 - \frac{p_{bc}}{1-p_{sc}} \right)^{n_{c}-1}}{p_{bc} - p_{sc} \left( 1 - (1-p_{sc})^n \right) \left( 1 - \frac{1-p_{bc}}{1-p_{sc}} \right)_{n_{c}-1}} = 0.089596 * p_{nn}
\]

independently on the distribution of the serial failures.

5. Concluding remarks

In this paper, we addressed one of the main causes of the lack of realism of most structural models for predicting the dependability of iterative software. The assumption of independence between the outcomes of successive executions, which is often false, may, in fact, lead to significant deviations of the result obtained from the real behaviour of the program under analysis. We have proposed a model in which both dependencies among input values of successive iterations and the effects of sequences of consecutive failures are taken into account in studying the dependability of iterative software. The effects of considering failure clusters, and the independence or correlation among successive inputs of the program have been analysed, first the effects of each in isolation and then together. The dependability attributes chosen for the analysis have been the probability of surviving missions (reliability after a certain number of executions) and the performability, representative of often conflicting requirements.
The proposed model can accommodate different distributions of the length of stays in failure regions. Therefore a number of distributions have been taken into consideration and their effects on the dependability figures analysed, in particular we used their probability of exceeding a given number of consecutive benign failures and their mean stay in failure regions. Two distributions representing the two extreme cases have been defined, which produce figures that bound those derived by all the analysed distributions.

References


Appendix: Approximate formula used for computing the performability

Due to problems with the tools used for obtaining the numerical evaluation we have not been able to compute the performability figures according with the general expression obtained in section 3.4.1. We therefore used an approximation obtained as follows. We observed that, for the distribution \( p^* \),

\[
P(\#\text{crossed regions} = i \land \text{mission success}) = \binom{n}{i} * (p_{sb} * (1 - p_{nn}))^i * (1 - p_{sc} - p_{sb})^{n-i}.
\]
p* maximises the number of failure regions encountered in a mission; we used this probability to replace the corresponding event in the general expression for the performability and we obtained the following approximation that is a lower bound for the general one:

\[
E[M_n] = P(\text{mission success}) \cdot n - \sum_{i=0}^{n-1} \binom{n}{i} \cdot \left( (1-p_{nn})^i \cdot (1-p_{sc} - p_{sb})^{n-i} \right) \cdot \frac{p_{ph} \cdot p_{bb}^{h-1}}{\left( \sum_{h=1}^{n-1} p_{ph} \cdot p_{bb}^{h-1} \right)}
\]

It is easy to see that the computed value for the performability is actually a lower bound because the new term we substituted in the general formula is bigger than the original one.

Observing that \((1-p_{nn})^i = \left( \sum_{h=1}^{n-1} p_{ph} \right)^i \cdot \prod_{k=1}^{i} p_{j_k} \cdot \left( \frac{1-p_{bc}}{1-p_{sc} - p_{sb}} \right)^{j-i} \leq 1\) and \(\binom{n}{i} = \frac{n^{(n-(j-i))}}{i^{(n-j+i)}}\), we obtain:

\[
\sum_{i=0}^{n} \binom{n}{i} \cdot p_{sb}^i \cdot (1-p_{sc} - p_{sb})^{n-i} \cdot \prod_{k=1}^{i} p_{j_k} \geq \sum_{i=0}^{n} \binom{n}{i} \cdot p_{sb}^i \cdot (1-p_{sc} - p_{sb})^{n-i} \cdot \prod_{j=1}^{i} p_{j_k}
\]

The difference between the approximated value of the performability obtained using this expression and the one obtained with the general formula in our setting of parameters is less than:

\[
\frac{1}{1-p_{nn}} \cdot \sum_{h=1}^{n-1} h \cdot p_{ph} \cdot \sum_{i=0}^{n-2} \binom{n-2}{i} \cdot p_{sb}^i \cdot (1-p_{nn})^i \cdot (1-p_{sc} - p_{sb})^{n-i} \leq (n-1) \cdot \frac{1}{1-p_{nn}} \cdot \sum_{h=0}^{n-2} \sum_{i=0}^{n-2} \binom{n-2}{i} \cdot (1-p_{nn})^i \cdot (1-p_{sc} - p_{sb})^{n-i}
\]

which is absolutely negligible considering that the values of the performability obtained in Figure 3 and Figure 4 are of the order of 5 \(10^5\).