

Statistical Testing: Qualitative Arguments for a Quantitative Outcome?

Martin Rendell and Fred Pollard

Rolls-Royce SMR, United Kingdom

Abstract

Statistical testing is a technique accepted by functional safety standard BS EN 61508-2:2010 for higher integrity software-based systems, either as an alternative to black-box testing and field experience or as a technique to support code verification. It has historically been seen as relevant good practice in the UK nuclear industry as a post-validation activity. Its primary usage has been to provide evidence to support an upper bound for the numerical reliability claim on a programmable system. Using past published examples of statistical testing and the authors' own experience of implementing statistical testing systems, this paper evaluates the practicability of using this approach for complex systems like reactor protection systems, covering each of the core conditions underpinning the execution of the technique (accurate test operating environment, test-case independence, determination of pass/fail and zero test failures). We examine divergence from these assumptions, and how this may undermine the quantitative claims of the technique. Additionally, we discuss the validity of the test outcome for a complex system, and in particular whether statistical testing's implicit model of a software-based system reflects what we know about software-involved faults due to its focus on system reliability, and not on reliability of safety functions. We conclude that while concepts and approaches used to implement statistical testing can provide useful qualitative evidence, statistical testing itself should not be considered a practicable method for obtaining high-confidence quantitative reliability estimates for complex systems like reactor protection systems, and as a result the benefits of other techniques should be considered.

1 Introduction

1.1 Purpose

This paper reviews statistical testing as a means of quantitatively supporting a reliability claim at a target statistical confidence level for a complex programmable system such as a reactor protection system¹, through literature review and the experiences of the authors. This review evaluates:

¹ For the purposes of this paper, we consider a complex system to be a system that implements or is involved in the implementation of many distinct safety functions. We use the definition of a safety function provided by BS EN 61508-4:2010: “*function to be implemented by an electrical/electronic/programmable electronic safety-related system or other risk reduction measures, that is intended to achieve or maintain a safe state for the equipment under control, in respect of a specific hazardous event*”.

- Whether it is reasonably practicable to achieve all the core conditions of statistical testing to an adequate level of rigour, and
- Whether it is justifiable to claim the targeted quantitative results for a complex system like a reactor protection system.

This paper explores several aspects of statistical testing that have previously been identified to be problematic by existing literature (e.g. Littlewood and Strigini (1993), Butler and Finelli (1993), Ladkin and Littlewood (2016), and Daniels and Tudor (2022)) and expands upon them based on experience and further research, and complements this with new contributions — in particular:

- Whether statistical testing provides confidence in safety-function reliability, which is typically what is of prime interest to plant design and the safety case;
- Practitioner observations of oracle and harness design, and the ramifications of inadequately conceived inter-test state-reset design; and
- Observations on the use and practicability of statistical testing for complex modern safety systems.

This paper refers to “software”, but most discussion is equally applicable to hardware description language (HDL)-programmed devices such as field-programmable gate arrays.

1.2 Overview of Statistical Testing

The input space and internal state of a software-based control and instrumentation (C&I) system is likely large, such that exhaustive testing is intractable for all but the simplest of systems.

Statistical testing of programmable systems is considered by BS EN 61508:2010 Parts 2 and 3 (referred to as “*probabilistic testing*” in Part 3) as a means of supporting a reliability claim on a software-based system in a specific configuration for a particular application, and has been promoted in the UK nuclear industry since the 1990s. BS EN 61508-2:2010 grades the technique’s recommendation level and technique effectiveness according to Safety Integrity Level. Proponents of statistical testing argue that it is a useful technique to support the qualification of a software-based system, on the basis that there are no other agreed methods of quantitatively estimating the reliability of such a system in advance of its operation, e.g. Kuball *et alia* (2004), other than through analysis of prior-use data if the system already exists, as described in BS EN 61508-7:2010.

A statistical testing suite subjects the test object to a series of demands, statistically representative of the expected operational profile that the real system will experience in its target application. In the case of a reactor protection system this demand could be a reactor plant fault that requires the protection system to take a series of actions to protect the plant (such as tripping the reactor and subsequently executing other safety functions to bring the plant back to a safe state).

Given N passed tests and zero failed tests (where a “pass” is when the test object has performed the appropriate safety function(s) within defined accuracy and timing tolerances), statistical testing theory allows for a claim of a probability of failure on demand (pfd) of θ

to statistical confidence α . This relationship is given by Equation (1), with typical numbers provided in Table 1 overleaf².

$$N = \frac{\ln(1 - \alpha)}{\ln(1 - \theta)} \quad (1)$$

Table 1 ~ Statistical Test Suite Size

α	N	
	$\theta = 10^{-x}$	$\theta = 10^{-4}$
50%	$0.69/\theta$	6,931
90%	$2.3/\theta$	23,026
95%	$3.0/\theta$	29,957
99%	$4.6/\theta$	46,052

Note that Equation (1) is derived from a frequentist approach, which deals with aleatory uncertainty on the condition of model assumptions being true. Some discussion in this paper uses Bayesian statistics instead, where epistemic uncertainties in the test arrangements can be factored in. Regardless, the discussions in this paper are ultimately applicable to both, though in some cases in different ways.

Statistical testing has been used in the UK nuclear industry as a means of quantitatively demonstrating confidence in the system's reliability. Statistical testing is recommended by NS-TAST-GD-046, the UK Office for Nuclear Regulation's (ONR's) guidance to its own inspectors to demonstrate software-based systems reliability (ONR 2023).

This paper distinguishes between statistical testing used to support a numerical reliability claim and operational-profile testing used as a qualitative test activity.

1.3 Structure of This Paper

This paper provides commentary on statistical testing across the following main sections:

- Section 2 summarises a review of literature on uses of statistical testing in industry.
- Section 3 looks at the challenges involved in implementing a statistical testing campaign in such a way that it supports the core conditions of statistical testing.
- Section 4 examines how statistical testing relates to the plant safety case, and whether the technique's relationship with system integrity and safety-function reliability is appropriate.
- Section 5 considers whether aspects of statistical-testing methodology are worth applying in non-statistical contexts, drawing on published case studies of successes from statistical-testing campaigns.
- Section 6 provides concluding remarks.
- Section 7 provides brief thoughts on future work as a result of this paper's conclusions regarding alternative approaches and techniques.

² Equation (1) is the classical frequentist single-sided bound for N independently and identically distributed Bernoulli trials. A Bayesian formulation also exists, which in practical terms provides very similar numerical results in terms of the relationship between confidence, reliability and number of tests.

2 Statistical Testing in Industry

In the UK nuclear industry, statistical testing has been applied and completed for different systems and components. This includes the following examples:

- Gough and Kuball (2014) describe execution of statistical testing by EDF Energy on a subset of functionality on the replacement Dungeness B Data Processing and Control System (DPCS). This comprised 460 statistical tests on the premise that it would demonstrate 99% confidence that the alarm functionality would meet its 10^{-2} pfd reliability claim, plus 26 “padding” tests to provide additional coverage of under-tested scenarios.
- Kuball *et alia* (2004) describe a research project running a form of statistical testing on a programmable logic system, which revealed a non-safety-significant defect that other test techniques in the same study had not detected.
- EDF Energy performed statistical testing on several smart devices and programmable logic systems, such as a motor protection relay, a pressure relief valve controller, a system comprising two devices providing spine break detection, and several conductivity monitors (Kuball 2024) (Walker 2024). The first three were performed as part of activities to support a claim of 10^{-2} pfd; the conductivity monitor testing was for a claim of 10^{-3} pfd (after 2004 voting³).

No complex programmable protection system has been subjected to statistical testing as part of nuclear-site licensing activities in the UK. There are two programmable protection systems to consider in relation to this claim: the Sizewell B Primary Protection System (PPS) and the Protection System (PS) at Hinkley Point C.

In the case of the Sizewell B PPS, the nuclear site licence was granted on condition of a successful campaign of dynamic testing, not statistical testing (Hunns and Wainwright 1992) (May et al. 1995). While the for-licensing suite size of 50,000 tests correlates with Table 1 (i.e. 99% statistical confidence in 10^{-4} pfd), the Nuclear Installations Inspectorate (NII, the contemporary nuclear regulator) determined that “*the test results could not be used in this way because the tests were not conducted according to the necessary rules of statistical experiments*” (May et al. 1995). Note that the operational profile used for the tests was restricted to infrequent plant faults, as these were the faults where the laddic-based secondary protection system was not claimed (Tooley and Stokes 1992). After licensing, the owning utility company “volunteered” to perform an additional 5,000-test statistically representative test suite, claiming a resultant ~99% confidence in a 10^{-3} pfd claim (Littlewood and Wright 1995) (Littlewood and Wright 1997). Note that it is not clear from available information how this claim is supported by the test-suite construction, which was designed to show the PPS’s reliability for fault scenarios for which functions were required to act with 10^{-4} pfd, i.e. no secondary protection. Furthermore, evidence was not found that all prior NII objections preventing statistical claims were resolved in the statistical experiments. In particular, in the description by Tooley and Stokes (1992) the eleven test scenarios in the initial campaign were focussed on a subset of functions, implying that the operational profile was incomplete. Tooley and Stokes (1992) argued (in the context of the non-statistical testing that they describe) that this was acceptable as the majority of other functions were tested by these scenarios anyway, but this is irrelevant for a later statistical testing campaign, where there is a need for coverage of the true operational profile as part

³ Two out of four

of the statistical core conditions⁴. May *et alia* (1995) do not indicate that the statistical testing experiments would include any more scenarios than the original dynamic-test campaign.

The Generic Design Acceptance (GDA) submission for Hinkley Point C (HPC) committed to a statistical testing campaign of 50,000 tests to support the 10^{-4} pfd claim on the PS (ONR 2012). However, this campaign has not yet been executed due to the development lifecycle stage of the plant⁵. Walker (2024) indicates a shift from a pure statistical testing campaign to a more hybrid approach with redistribution of test cases within the test campaign. Other prospective nuclear new-build licensees in the UK, such as Horizon Nuclear Power's UK Advanced Boiling Water Reactor (ONR 2017) and Westinghouse's AP-1000 (ONR 2011a), also proposed statistical testing of their respective protection systems in their GDA submissions, but these new-build projects were shuttered before work could progress beyond desktop-based feasibility studies.

Outside of the UK nuclear industry, there is less published evidence for the implementation of statistical testing. During research undertaken for this study, limited examples were found in the public domain; the most well-described was statistical testing of the trip functionality within the loop-operating control system of an Advanced Test Reactor experimental loop at Idaho National Laboratory (US NRC 2017). In that paper, the interpretation of the test results provided an illustration of the epistemic uncertainty in statistical-testing outcomes⁶. “*Statistically valid, trajectory-based random testing*” was described as necessary for the Canadian CANDU shutdown system software (Ichiyen 1998), though further detail is not available. “*Reliability qualification*” is described in a CANDU software engineering standard (Ontario Power Generation Inc. 2013), including reference to key requirements in statistical testing. A guideline document (Strigini and Littlewood 1997) for application of statistical testing indicates that the European Space Agency was interested in statistical testing in the 1990s, but the extent of its application is not known to these authors.

There is mention of “*reliability demonstration tests*” including for software-based systems in the non-nuclear standard IEEE 1633-2016, though this comes from a commercial-product testing perspective that balances customer and manufacturer risk and looks to identify defects likely to be revealed in operation and/or to calculate a mean-time-before-failure (rather than pfd) measure. IEEE 352-2016 describes “*acceptance sampling*”, though based on the sample figures used it does not seem to consider it for numerical targets more onerous than 5×10^{-3} ; the descriptive text refers multiple times to “*quality control*” rather than particular safety-context intent. Relatedly, the US Nuclear Regulatory Commission (NRC) regulatory structure “*relies on the use of a qualitative approach for determining software reliability*” (US NRC 2022). Many other standards across numerous industries do not attempt to quantify the contribution of software to the overall reliability of a system. For example, in general functional safety (e.g. IEC 61508) it is generally accepted that quantified reliability relates to random hardware failures and is distinct from systematic safety integrity, which “*cannot usually be quantified*”. Other industrial standards such as aerospace

⁴ Note if the contents of Section 4, where we discuss whether we are testing systems or functions, is considered here then the situation is worse — the “collaterally tested” functions are tested using only a subset of the operational profile scenarios relevant to them.

⁵ As of 2024, the predicted cost is in the millions of pounds, and the project is still several years away from completing to support fuel-load (Walker 2024).

⁶ A test suite of 10,000 tests was performed to provide evidence to support a 10^{-4} pfd claim on the software's trip functionality, where 5th (5×10^{-6} pfd) and 95th (3×10^{-4} pfd) percentile estimates for the pfd were calculated. These figures are based on a uniform prior, zero observed failures across the 10,000 tests, giving a posterior distribution of Beta(1, 10,001).

(e.g. SAE ARP4754A, RTCA/DO-178C), rail (e.g. EN 50129, EN 50716), and automotive (e.g. ISO 26262) similarly separate out quantified reliability from systematic failures from elements such as software, controlling the latter via process controls. This is in contrast to the UK nuclear approach, which arose in the late 1980s and early 1990s when most such standards did not exist, and which has generally required quantification of safety function reliability in support of demonstrating sufficient reduction of risk.

There are no other recent statistical-testing projects (in the sense discussed in this paper) identified by this literature review, though an International Atomic Energy Agency (IAEA) paper suggests that the technique is “*often*” used. IAEA (1999) lists statistical testing as an “*optional*” technique. A regulatory common-position paper (ONR 2024) notes that statistical testing can be used either as a confidence-check during validation (i.e. as per ONR (2023)) or as an activity during the development lifecycle’s verification (akin to Musa (1993)).

All examples of statistical testing identified in this review are believed to have been performed in an open-loop approach⁷. No evidence was found for closed-loop statistical testing of large software-based systems during this review; for example, Sizewell B PPS (May et al. 1995), Dungeness B DPCS (Gough and Kuball 2014), and Hinkley Point C PS (ONR 2011b) are all described or implied in open-loop terms, i.e. pre-defined input transients for each test case⁸. The comparison between open- and closed-loop is beyond the scope of this paper, but warrants consideration when building a statistical test harness⁹.

3 Challenges of Implementing Statistical Testing

3.1 Core Conditions

Existing literature places the following *core conditions* on the implementation of a statistical testing campaign:

1. Each test case shall be selected from the (known or predicted) operational profile of the real system, such that the number of test cases based on each operational-profile scenario is proportional to the frequency of that scenario in the operational profile.
2. The test object’s behaviour shall only be dependent on the stimuli received in relation to the current test case; i.e. previous tests shall not affect the results of the current test case.
3. The test object shall use the same hardware and software version/configuration as the real system.

⁷ Open-loop testing: the test object’s behaviours have no effect on test-case stimuli presented to it, usually because the test-case data is generated in advance of test execution.

⁸ Closed-loop testing: the test object’s outputs are fed back into a live simulation of its environment, such that the following time steps may be influenced by the test object’s behaviours. This approach can present additional challenges for test-harness integration, but can provide more powerful evidence (see Section 6).

⁹ Various studies (for example Leveson (2004), UK HSE (2003), EPRI (2008), and Prokop (2024)), have noted that, typically in dangerous behaviours of a system, every component in the system operated exactly as intended. A study of space systems failures since Ariane 5 observed that standard V&V processes are generally effective at avoiding errors in translation from requirements to code that cause failures, whereas errors in data and in particular specifications were the cause of failures studied (ASCA 2007).

4. There shall be a means of checking whether the test object has provided the correct response to each test case.
5. In order to estimate statistical confidence α and probability of failure on demand θ , there shall be no observed failure by the test object to correctly respond to a demand.

These core conditions contribute to the justification that the statistical assumption of independent and identically distributed (i.i.d.) demand-sampling from the real operational profile is achieved.

Note that core condition 3 (same software/hardware version) is effectively derived from core condition 1 (accurate operational profile) because the software's environment includes its system hardware and the behaviour of each part of software. They are treated separately in this review as they can be encapsulated neatly as separate subtopics, though often in literature the "same software/hardware version" condition is not enumerated explicitly, e.g. see US NRC (2026).

Adequately achieving all five core conditions is especially challenging for large, bespoke systems such as a reactor protection system and for systems with high reliability targets. BS EN 61508-7:2010 explicitly notes that "*in practice, it is very difficult to demonstrate ultra-high levels of reliability using these techniques*". IAEA (1999) states that:

...the production of the test data and the evaluation of the correct results become an enormous and demanding task. The effort to establish the part of the input space to be tested and the distribution of test data is also very large.

The following subsections consider each of these core conditions and the challenges associated with implementing them. These subsections highlight some of the ways in which it is difficult or impossible to quantitatively support the core conditions, which undermines the ability to make quantifiably confident claims of the pfd. An inability to adequately meet all these core conditions makes it difficult to justify the i.i.d. Bernoulli demand models underpinning Equation (1).

Figure 1, overleaf, shows the key constituent parts of a test system, all of which need to be adequately designed, built and used in such a way as to fulfil the core conditions. A weakness in any of them may result in an unquantifiable effect on the outcome, if that weakness is even identified by the test engineer at all¹⁰. Inadequacies may affect the statistical conclusions' veracity (e.g. a defective oracle does not detect a fault) or validity (e.g. the test operational profile omits key parts of the real operational profile).

While the following terms can be used interchangeably to some extent, this paper uses the following definitions:

Initiating Event: For a test suite for demand-based functions, this is the event that marks the beginning of the demand on the system's functions. For a reactor protection system, this might be a pipe break in the reactor primary circuit.

Scenario: A definition of the progression of the plant properties following an initiating event. This can be used as the basis for test cases.

Test case: An instance of a scenario, defining a series of inputs to the test object.

¹⁰ The "simulated subsystems" box in the Figure refers to where parts of the C&I system are not directly tested, for example redundant divisions of a reactor protection system.

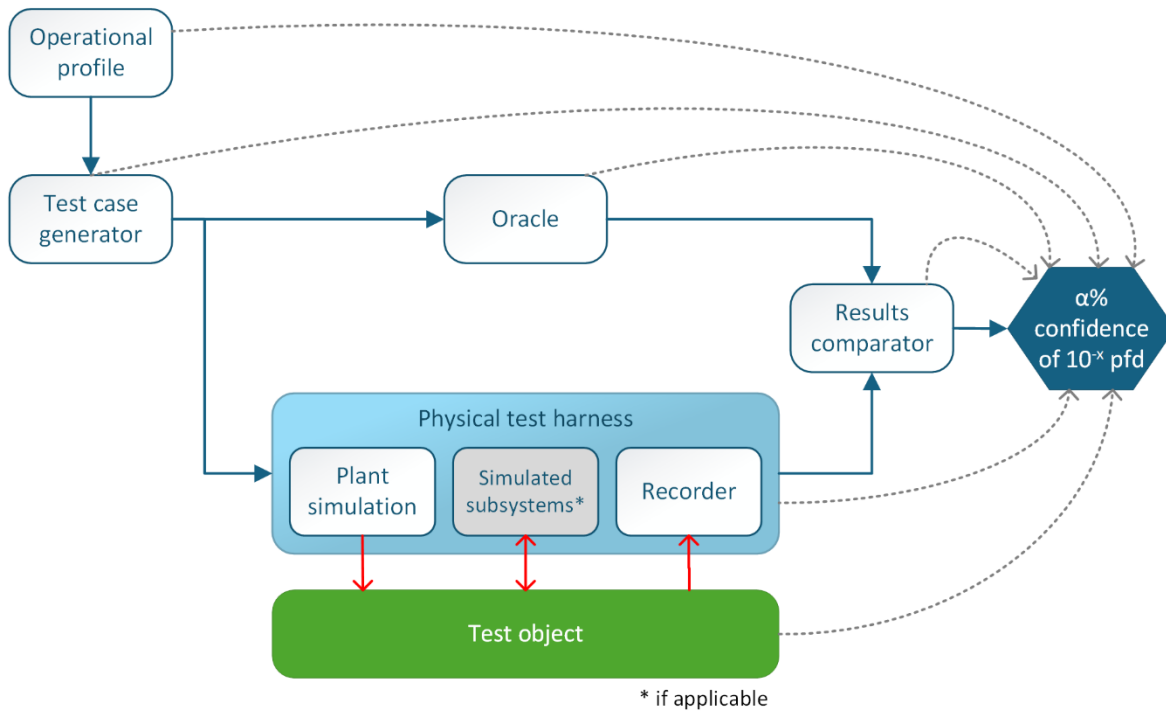


Figure 1 ~ All Statistical-Test Harness Components Contribute to the Veracity and Validity of the Statistical-Testing Outcome

3.2 Implementation Challenge 1: Building the Test Operational Profile

Establishing an accurate test operational profile is a significant undertaking¹¹. For a reactor protection system, there is an inherent need to model a large proportion of the plant. There may be many different plant fault scenarios¹². However, unlike in typical transient analysis, it would not be valid for a statistical testing campaign to only model the “bounding” plant faults, because there is no definable relationship between plant fault severity and likelihood of failure of a protection-system software function. The operational profile defines demands on the test object, but the demands may be affected by other plant systems (e.g. secondary protection systems, control and prevention systems), each of which may actuate or not actuate (either by design or due to a fault). This adds significant complexity. To achieve a well-detailed operational profile for a reactor protection system will require years of analysis effort, involving sensitivity analysis and determination of what can and cannot be modelled with the simulation tools¹³.

Can a probabilistic safety assessment’s conservative best-estimate (50% confidence) plant initiating-event frequencies, which are used to generate the operational profile distribution, adequately support a statistical-testing outcome of $\alpha = 99\%$? Bishop and Povyakalo (2017)

¹¹ This literature review revealed relatively emotive language being used to describe this task, such as “a task to reckon with” (Horgan and Mathur 1996) and “enormous and demanding task” (IAEA 1999).

¹² Walker (2024) states that Hinkley Point C’s PS statistical testing covers 150 separate initiating events across all plant states, covering ~70 unique design-basis initiating faults, across a wide range of initiating-event frequencies.

¹³ In fact, depending on the capabilities of the simulation tools, there are some fault scenarios that may not be possible to model in a way that is useful for the test operational profile; this would represent a gap which will need justification.

consider that the frequency of each scenario in the modelled operational profile may be inaccurate, and that this may affect the value of α . Bishop and Povyakalo demonstrate that extra tests can be added to the suite as “padding tests” to reach the original target α , but this has the undesirable effect of increasing the overall test-suite size.

Even if the relative frequencies of initiating events are all representative, the modelled operational profile can never truly be complete¹⁴. However, it seems reasonable to suspect that the areas of the system’s design that are more likely to contain flaws are the areas where a possible operational demand was not foreseen. This might be in terms of a plant or systems fault, or more likely a “random” operator or maintenance error or action on the plant unforeseen by the system design. Connected devices, e.g. sensors, may not behave exactly as simulated. Fault studies may consider the effect to be bounded by plant faults (e.g. an operator erroneously opening a valve may be equivalent to a pipe break) but can we be sure there is no difference in effect on the test object’s stimuli? Ageing, upgrades, and replacements in the plant will further perturb the true operational profile in ways that cannot be predicted before plant commissioning; arguably, the best we can claim is that the statistical-testing operational profile is accurate for the moment of plant switch-on only. Strigini and Littlewood (1997) acknowledge ageing effects, but do not provide a solution, even within the scope of their paper which is for smaller and less complex environments than a nuclear power plant. If the same system is to be installed in a different environment (e.g. a reactor protection system in a different plant that is not perfectly identical) then consideration is needed about whether the test operational profile is still valid.

Miranda and Bertolino (2017) sum up the challenge’s significance succinctly, with reference to Lyu (1996), when they say,

... operational profile-based testing is grounded on the notion that not all faults have the same importance: depending on how it will be exercised by the users, a program can show quite different levels of reliability.

Note that while Musa (1993) suggests that operational-profile strict accuracy is not necessary to gain improvements, this is in relation to operational-profile testing during system development for the purposes of detecting defects early, not for estimating high-integrity system reliability. This distinction is made clear by Strigini and Littlewood (1997), who note that:

... errors in the operational profile can in theory cause the estimation to be radically wrong (taking any value between 0 and 1). [...] There is insufficient experience for a tester to know in advance all the approximations that can be allowed without causing excessive error.

For a reactor protection system, it will be impossible to validate the test operational profile by comparing to the system once it is in operation, because its demand frequency is (hopefully) so low that there will be a minimal body of evidence, even using data from every reactor in the world (IRSN 2017). The more likely outcome of attempted validation would be to find that a real fault scenario diverges from what was modelled in the test scenario, due to any of the reasons discussed in this subsection. For non-statistical testing, divergence would just be a limitation of the simulation, but for statistical testing it undermines the validity of the statistical assumptions.

¹⁴ As noted by IRSN (2017), there are too many different input sequences to know which are likely, possible or not possible. BS EN 60880:2009 notes that the difficulty in defining the operational profile accurately increases with number of inputs.

As highlighted by Littlewood and Strigini (1993), reliability is a measure of a product working in a particular environment. We might be measuring the reliability of the system as a test object in the test laboratory, but there is a lot of work to show this environment allows the reliability result to be translated to reality.

3.3 Implementation Challenge 2: Ensuring Test-Case Non-Interaction

There are two main options for ensuring non-interacting test cases: 1) reset the test object (e.g. power cycle and return to a known initialised state), or 2) “flush” any internal state through carefully selected series of inputs over a defined period. Both approaches usually require a settling period to ensure the internal state has reached a stable, static and predictable state after the reset/flush.

When reinitialising the test object through either means, the re-initialised state must be understood. This requires assumptions of the internal workings of the system, which is at odds with the “black-box” intended nature of statistical testing, especially for more complex systems¹⁵. Kuball and May (2007) suggest an approach for clearing the internal history, though it is not clear how easily scalable this is to larger systems with interconnected logic (it is likely for example that a reactor protection system will have minutes or hours of history to clear), and may be susceptible to undocumented or otherwise unexpected or opaque internal history, which is more likely for more complex systems.

Two potential options for confirming history length are to perform static code analysis or to instrument the executable during testing (May et al. 1995). However, the static analysis may not be complete until after statistical testing needs to start (particularly challenging if test infrastructure changes are needed, or if the analysis reveals very long reset times). Instrumenting the executable under test (potentially a significant undertaking to perform exhaustively due to complexity) would violate the “same version” core condition of statistical testing.

This reset challenge is a reason why ONR’s (2023) guidance to inspectors recommends that systems are designed with statistical testing in mind; however, this can be challenging to achieve in practice, especially when building applications on pre-existing platforms, rather than purely bespoke systems. Strigini and Littlewood (1997) suggest methods for designing systems as such, though for a reactor protection system the limiting factor is likely to be the plant requirements (i.e. the application functionality needs to have timers ranging from seconds to hours). Ultimately, there is unlikely to be enough incentive for manufacturers to design their systems to support statistical testing directly, especially on platforms with significant existing history, and even with such functionality implemented, there would be a question of whether the statistical testing engineer can rely on this functionality or would have to confirm it independently.

With the *possible* exception of static analysis¹⁶, none of these reset approaches will definitely reveal defects in the system where internal states and memories do not behave in the expected manner, unless these defects are directly revealed by any pre-test checks or the test case itself. It could be argued that if the test case does not reveal a defect, then that defect has no functional effect, but this conclusion is dependent on the validity of the test case and does not acknowledge that we are still only sampling a tiny fraction of input space. Defects

¹⁵ This is one of the reasons statistical testing of non-simple systems usually ends up being “grey-box” testing in practice.

¹⁶ “Possible” because calculation of memory length for a system comprising concurrent subsystems may require very detailed static analysis with a variety of techniques.

due to long-term operation, e.g. counter overflow (EPRI 2010), or other background activities, e.g. exhausting memory (Coolen 2012), may not be detected during statistical testing, especially if the test object is physically reset between tests. Ladkin and Littlewood (2016) discuss masked dependencies that may not have been recognised; without extensive static analysis it may be difficult to prove that none of these affect reset and re-initialisation. Such defects would undermine the assumption that the system's response to a demand is independent of the time (and behaviours) between reset and demand.

Test-case independence (or at least control of side effects) is usually required by other testing techniques as well (not just statistical testing). However, the crucial distinction is that statistical testing is specifically trying to demonstrate the numerical probability of failure on demand, and so is reliant on test independence to support the quantitative outcome.

3.4 Implementation Challenge 3: Test Object and Real System Are Identical

The constraint of the test object being identical to the real system (as identified in ONR (2023) and Ontario Power Generation (2013)) exists because the hardware/software versions and their configuration effectively contribute to the operational environment of the software (see core condition #1). It helps to narrow down the state space that needs to be explored, since a single difference in configuration file or software logic could change the movement through the system's state space in unpredictable ways.

This system configuration constraint means that statistical testing for reliability determination cannot commence until after final validation of the system¹⁷. For a bespoke system, particularly something like a reactor protection system, this is likely to place statistical testing on the critical path for plant commissioning and operation, because the software is likely to be finalised close to plant operation dates in order to account for other plant changes on which the system is dependent. Considering that a test suite may take months or years to execute¹⁸, this places pressure on the statistical testing campaign to complete. Running multiple instances of the test setup in parallel (i.e. multiple test objects) would of course reduce this, but at significant expense due to the size of these highly specialised pieces of high-integrity equipment. Running simulated copies in parallel would undermine the same-configuration assumption.

Available literature (for example, Tooley and Stokes (1992), ONR (2012), ONR (2020) and ONR (2011a)) indicates that any statistical testing (or in the case of Sizewell B, dynamic testing) performed or planned to be performed on nuclear reactor protection systems focuses on a test object that is a copy of only a single physical division of the system (i.e. one of the several redundant copies of the same protection equipment and logic), with other redundant divisions simulated by the test harness. As noted in regulatory reports, this requires

¹⁷ See May *et alia* (1995) and Strigini and Littlewood (1997). Gough and Kuball (2014) argue for waiting for the final validated version from a practical perspective as well.

¹⁸ Tooley and Stokes (1992) indicates that in the initial version of the Sizewell B dynamic testing, each test took on average 3 minutes (whereas Littlewood *et alia* (1998) cite 5 minutes per test), and Dolan (1994) states that the ~50,000-test Sizewell B campaign took 6 months. We posit that the average-time difference may be due to different iterations of the test suite (perhaps with changes to testing approach), and note that 5 minutes per test aligns with 6 months test duration, if running tests "24/7". In modern protection systems, with increased autonomy and complexity, any means of accelerating or truncating test cases are less feasible. Gough and Kuball (2014) note that 486 transients (for a system with a 10^{-2} pfd claim, plus padding tests) averaged 4.75 hours, taking a total of 96 days. For a 46,000 test suite with average test length (including reset time) of 30 minutes, the campaign would need to run continuously for nearly 1,000 days. In addition to the nominal time to run the tests, there is usually a need for additional padding time to account for pauses in testing, e.g. for test system maintenance, problem-solving of the test setup or investigating oracle discrepancies (which for Sizewell B took considerable effort (Arthur 1993)).

justification¹⁹. Without adequate justification, the test suite size will need to be multiplied by the number of redundant divisions (this may be challenging if divisions are not using an identical binary and hardware configuration, which is likely).

There appears to be no conclusive research into using emulation or simulation of the test object to accelerate statistical testing, and we are not aware of any research or examples of a statistical testing campaign being performed with emulated test objects that has been able to quantitatively support the statistical testing theory convincingly without making qualitative arguments that move us away from the core conditions. May (2017) developed a theoretical approach for running a large number of tests on an emulated object and then a small number of confirmatory tests on a real test object, to support a quantitative estimation of α and θ . However, May's approach has not yet been developed further.

It seems inarguable that running tests in simulation will provide some qualitative confidence at least, even if it cannot formally support a quantitative confidence due to the hard-to-quantify assumptions needed. Since it is possible to run many more tests in simulation than on a real system, does a focus on calculating α distract us from taking steps to achieve more general benefits?

Another option for tackling the programme/schedule problem is to perform statistical testing on a version of the system that precedes the final validated version. This violates the core condition of testing the same version as in operation, so the value of the testing is diminished. A successful pre-final version statistical testing campaign will give some confidence that there is a good base level of reliability, but a single defect fix or hardware/software change made for the final version could introduce (or unmask) a defect, thus changing the system's reliability "*in subtle but unknown ways*" (Strigini and Littlewood 1997). It may be possible to address this by running (on the final version) a second full campaign or a reduced-size campaign. The reduced-size campaign is likely to need to be very short to allow completion prior to operation, and a full re-test would have to be finished after the plant is operational unless the operation date is delayed, so there is a risk that we would only discover that the final protection system's reliability is less than expected once it is already in use²⁰. A partial re-test would be challenging to justify quantitatively, since it would need to make further assumptions on the homogeneity of the software (or alternatively open the black box to work out the validity of assumptions, and thus tie our understanding of the software behaviour to its implementation characteristics rather than purely to its response to test trajectories). We assert that a partial re-test would be a qualitative demonstration only; but if a partial re-test is a qualitative demonstration, what was the point of the pre-final test having a quantitative goal? Could we have made better use of test time on the pre-final version if we were not chasing quantitative outcomes?

As discussed in Sub-section 3.6, the need to use the final version results in additional project schedule risks — the testing would be performed at the end of the project (after the development lifecycle has finished) so the onerous requirements of statistical testing in response to test failures would have an even more severe effect.

¹⁹ In part because protection systems' divisions' application logic are not usually completely identical, e.g. ONR (2011a). Inter-division communications will need to be realistically modelled, and differences between the object code of the real division and the object code of the simulated divisions, and the resulting timing behaviours, may exist.

²⁰ This becomes both a safety risk (fuel is present in the core and we have now identified a defect) and a reputational risk (we have turned the system on before running all the tests). The reputational risk aspect relating to protection-system testing is one known all too well to Sizewell B, where its initial 50,000 test run showed ~50% discrepancies. These discrepancies were all demonstrated to be weaknesses of the test harness, but the numbers were obtained by the British press and sparked public concern, even leading to a motion in the UK Houses of Parliament (Hansard 1993).

Note that the discussion around this implementation challenge has focussed on the software version of the real and test objects, though the same stipulation also should apply to the hardware. It is not inconceivable (and indeed not quantifiable with respect to the statistical-testing confidence level) that minor differences between the hardware in the test object and the hardware in the real system may affect the software behaviour. Further, the effect of differences in local electromagnetic interference may change inputs to the software. While potentially negligible, this is just another example of where we have to make some qualitative judgement on how representative our test setup is if we want to support the statistical-testing quantitative outcome — and, for examples such as those discussed in this subsection, there is no evident basis for quantifying the relationship between such differences and the claimed confidence level.

3.5 Implementation Challenge 4: Determining Pass/Fail (Detecting All Faults)

When statistical testing is carried out on a system that performs one or more safety functions, what should the acceptance criteria for a test be? In statistical testing, it is that the test object has performed the correct actions at the correct time, where “correct” is in reference to the system requirements specification for the system under test. Tolerances are allowed for the timing of the actions, because of real-world delays like propagation of signals through asynchronous components, sample rates, and so on. No tests shall have unexplained failures if the statistical confidence in failure rate is to be achieved.

Determining “correct” traditionally involves construction of an “oracle” tool by the statistical-testing team, which models the system requirements independently of the test object, much like in other forms of testing. This can be a significant undertaking especially for more complex systems.

While the oracle should not use development artefacts from any software/hardware created after the finalisation of the system requirements, in reality the degree of independence from the implemented system often ends up being weakened slightly so that the oracle can account for how the implemented system actually behaves (e.g. time response, algorithm choices). A purely black-box approach to building the oracle is often not possible when accuracy and precision need to be high (e.g. when testing safety-critical systems), and so diligence is required to ensure that common-mode failure is not established between the oracle and test object as a result of “peeking” at test-object design data — such failures may lead to false negatives.

On the other hand, an oracle that is low-fidelity, or overly stringent, can lead to many false positives (tests marked as fail that should have passed), which can significantly increase the manual workload to show that the test object is not to blame, adding effort at the tail end of a test campaign. For example, Sizewell B’s statistical testing saw ~50% of its initial ~50,000 dynamic tests marked as “discrepant” by its oracle (Dolan 1994) — the majority of these were due to oracle shortcomings²¹, and all were shown to be false positives (Arthur 1993). While oracles like this are now better understood, and so likely to have a better accuracy, they can pose a risk if inadequately conceived or constructed. Building an oracle is hard, as it requires implementation of the system requirements in an ideal model, while also accounting for behaviours of both the physical test harness and the system under test. If redundant divisions are simulated as part of the harness, this can add further complexity.

²¹ Though many were due to other aspects of the test harness, e.g. the electrical simulation and monitoring.

Ladkin and Littlewood (2016) look at this implementation challenge from a different perspective. If a test fails because of defect *A*, then the conditions required to activate defect *B* may not be reached in the test. As a result, we may fix defect *A*, but defect *B* would only be revealed on the second run (if at all, as it could be in an uncommonly exercised part of state space). Linking to implementation challenge #5, this may place severe pressure on the test schedule.

Littlewood and Wright (2007) use a simplified model of a test campaign as an input to a Bayesian belief net, and find that the more test cases that are successfully executed with no failures, the greater the likelihood that the oracle is passing tests without justification (i.e. the oracle is failing to detect or masking genuine defects in the test object) and the greater the likelihood that the specification is wrong, and so the lower the confidence in your statistical testing result. The paper suggests this result is unintuitive, but it is not inconceivable that an oracle may be unintentionally built to be lenient on the test object due to its design being influenced by the implementation under test. In reality, to reduce false positives, trial campaigns might be run as part of the test system's acceptance testing — this requires rigour to ensure that defects in the test object are not falsely attributed to issues in the test infrastructure.

Knight and Leveson (1986) conclude that n-version programming does not guarantee absence of common failures. Analogously, consider that the test object and the oracle (including test harness) are two diverse systems implementing the same functions. In this case, common-cause failure is possible, and indeed if one of the diverse systems is less reliable than the other (i.e. the oracle, which is not high-integrity software), then its own reliability will dominate the overall probability of the two systems coincidentally failing to actuate a safety function. Coincidental failure of the oracle and the test object to actuate a safety function at the right time may result in the test passing instead of failing (false negative). In a system with many safety functions, this false negative may be missed by the test engineer due to the correct concurrent actuation of many other safety functions — i.e. the test engineer sees that there was some safety response, but does not see that all necessary safety responses occurred.

In summary, because the oracle is software-based, its own reliability is not quantifiable. If we cannot guarantee a perfect oracle, or quantify its reliability, how do we support a 99% confidence calculation in the reliability of the system under test?

3.6 Implementation Challenge 5: No Tests Shall Fail

Why is this a challenge? Surely this is the ultimate goal of any test suite? This subsection highlights how a test failure may challenge the project schedule, where such a failure would leave the accumulated evidence, and whether statistical testing is the optimal use of effort.

If a failure is detected during statistical testing, then, unlike in other forms of testing, it is challenging, or maybe impossible, to use impact analysis to understand how the defect fix affects the test outcome, i.e. the target confidence in reliability. As a result, after the fix a full suite re-run may be needed, or at least a significant expansion of the test-suite size. This is clearly a concern for the project schedule, because the testing should be performed on the final version and so is near the critical path. This creates friction with the safety justification — achieving the aim of demonstrating confidence will directly lead to operational delays.

In addition, potentially unintuitive statistical justifications may be needed for any course of action other than full re-test, which is undesirable in a safety-related justification²².

Any test technique that encounters a failure will likely require additional subsequent effort, but the distinguishing feature of statistical testing is that the design of the test suite is focussed on statistical distribution, not on, for example, functional or input-condition coverage. Therefore, if a test failure invalidates the statistical value of the suite, then the evidence generated by the suite until that point is likely to be less compelling than that which other techniques may have generated in the same time period. This is especially the case in relation to testing performed on pre-final versions (e.g. simulation testing of the application prior to hardware integration) — potentially far higher test throughput can be achieved at that point, i.e. at a time that is easier to fix any defects (and indeed the fixes are likely to be more thorough, as late-stage fixes often may be constrained so as to reduce late-stage impacts).

Kuball *et alia* (2004) state that their statistical testing campaign revealed failures that other forms of testing in their study did not, but concede that the consequence of this was that a reliability figure could not be claimed from the statistical testing until the defects were removed and the system successfully re-tested. This was despite the fact that the defects did not affect safety. Equation (1) is blind to defect severity.

The critical consideration is whether test time and effort should be spent performing statistical testing (especially considering all of the challenges that have been identified in this review paper) rather than other techniques that forgo a quantitative demonstration of a reliability target in favour of more targeted and justifiable testing.

The challenges associated with understanding test value when there has been a failure links to Section 4, where we consider what statistical testing tells us about the reliability of individual safety functions.

3.7 Other Implementation Challenges

The above subsections have covered several of the assumptions and constraints in Figure 1, but not all. It is likely that, as with many test systems, the test harness's injection of signals into the test object will be only a model of how real signals are injected onto the real system — for example, instead of a continuum of inputs, it is likely that signals will be injected in a regular series of discrete time steps, likely with each signal being injected synchronously. This reduces the representativeness of the stimuli and may mask temporal-based defects in the test object.

As already noted, test-suite execution length usually presents a challenge in statistical testing. In high-integrity validated systems, where statistical testing is typically used, faults are rarely detected, so there is little data available to indicate at what point in a test campaign this detection will occur: are faults first detected in the first few iterations of a particular scenario, or do most defects require such specific conditions that they are likely to need many tests to be found? If the former, this makes it harder to justify to programme managers why statistical testing should continue for 5,000 or 50,000 tests, when defects are likely to be found in the first few hundred tests²³.

²² Responding to a statistical-testing failure is the subject of several papers (Littlewood and Wright 1995) (Littlewood and Wright 1997) (Korhonen et al. 1997) (Strigini and Littlewood 1997).

²³ Indeed, to de-risk the programme, a prudent statistical test campaign would seek to run tests based on each scenario at the beginning to catch immediate failures, and only then start to do repeats (variations) of each scenario.

The idea of uniformity of quality across the software under test (which we perceive to be an implicit assumption of statistical testing when performed at system- rather than function-level) may be challenged if, during the development lifecycle of the test system (including harness, test case generator, etc.), some trial tests are run (not necessarily with a statistical distribution) on the test object (possibly a pre-release version) using the test system. A test harness for a complex test object is challenging to build, and likely to have flaws in its interface with the test object, so this trial approach may be essential for a practicable test project. If a fault is detected in the test object during these trials, then the safe thing to do would be to fix it before the final version. But any such faults will make it even more important that the test operational profile is complete and representative, because now we have only improved the code quality in the area of code that our test profile will test. Ultimately, we are trialling our test system's correctness using the test object it is built to test. If we do a formal campaign and achieve 50,000 failure-free tests, is this because the test object is highly reliable, or is it because the trial tests drawn from the test operational profile have already detected all the faults that can be revealed by the test operational profile? This may qualitatively make sense (we have improved quality where we think it is likely to matter, as per Musa's (1993) approach) but it may make it harder to justify the validity of the quantitative calculation of α . It also places more importance on our test operational profile being complete compared to the real operational profile throughout the operational life.

In this Section, Bayesian inference techniques have already been touched on in a specific case (the discussion about Littlewood and Wright's (2007) result indicating that continuing to show test passes may call the oracle into question). A more recent paper (Salako and Zhao 2023) uses conservative Bayesian inference (CBI) to examine whether the i.i.d. criterion for sampled tests is necessary for a statistical testing result, and argues that it is not. However, while it proposes an approach that does not require i.i.d. sampling in a test suite, the high levels of confidence achievable if we did assume i.i.d. sampling are unobtainable in their conservative Bayesian approach. Salako and Zhao find that in fact unless the engineer has very good confidence $1 - \phi_2$ that tests are not positively correlated (i.e. the engineer has confidence that the success of one test does not imply the success of the next; this correlation might indicate that for example the tests are not challenging enough, the oracle is not good enough, or that state is not reset between tests), then confidence will reach a peak during a campaign if all tests are passing, and then diminish with further successful tests. In the extreme case of low confidence in no correlation then running any tests immediately diminishes the confidence established from prior knowledge right from the start of the campaign (see Salako and Zhao's Figures 4e and 4f for $\phi_2 = 0.3$; Figure 4e is reproduced overleaf as Figure 2, where $Beta(1, 10,000)$ represents a prior representing confidence from other sources (e.g. development quality, analysis, other test) of 10^{-4} pfd)²⁴.

On the face of it, Salako and Zhao's work seems to enhance the understanding of statistical testing in a pragmatic and conservative manner, providing a grounding for interpreting test results. However, we consider running a test campaign using this CBI approach:

1. The engineer designs a test system and tries to minimise ϕ_1 and ϕ_2 through good test-system design and implementation²⁵.
2. The engineer starts running the test suite, running 10,000 tests initially.

²⁴ Salako and Zhao (2023) Figure 4f uses a prior with lower prior confidence in the test object's pfd, showing that achievable confidence from this testing is even lower and drops off more sharply.

²⁵ ϕ_1 is the confidence that tests are negatively correlated.

3. Zero test failures due to the test object are encountered (i.e. no false positives identified).

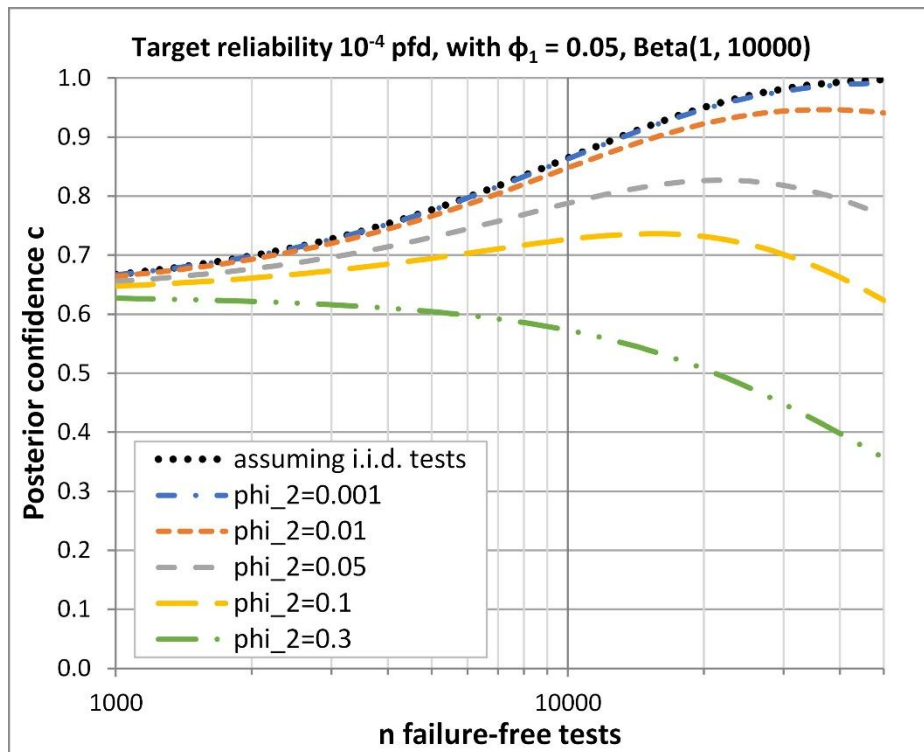


Figure 2 ~ As Confidence in Test Setup Diminishes (Higher Φ_2), the Lower the Achievable Confidence in Test-Object pfd (recreated from Salako and Zhao (2023))

Using Figure 2, what confidence do we now have in the pfd? If we have high confidence that our test setup is good and tests are not being unjustifiably passed, then we should keep testing to increase confidence further. If, however, our confidence in our test setup is lower, any more tests may reduce our confidence in the pfd (that is, if our confidence has not already started reducing already).

With this CBI approach, we have gone from the traditional statistical testing approach of assuming our good test-setup design fully meets the core conditions and thus we use Equation (1), to a situation where we assume that our ability to quantitatively define our numerical confidence in properties of the outcome of the test campaign (i.e. test correlation) is strong. From all of the implementation challenges discussed already in our paper, it is unclear how the test engineer could determine whether their ϕ_2 was 0.001, 0.01, 0.1, 0.3 or any other value with enough certainty to be of use.

We argue that use of CBI does not solve the underlying challenge discussed throughout this paper. Even with the additional mathematics of CBI, we are just using qualitative, subjective judgement to underpin quantitative parameters. The research by Salako and Zhao (and others) into CBI exposes mathematically (and graphically) the consequences of over-trusting a test setup, but does not provide a means of dealing with the test engineer's quantitative judgement being inaccurate. Are we trying to do what humans are notoriously bad at: estimating confidence and probabilities based on judgement alone?

One could adjust where the tipping point of the confidence curve is by adjusting the parameters ϕ_1 and ϕ_2 , but this is what we tried to do at the start by doing good test design. After those 10,000 tests in our hypothetical test campaign, we are only now further questioning our test design because we have passed all the tests. If we do take time to improve the test design, and then continue testing, how do we quantify the confidence-building value of the first batch of tests?

To draw a positive note from the CBI work, one may argue that there is indeed a justification for performing smaller statistical-testing campaigns than is traditionally done — for the 10^{-4} pfd system in Salako and Zhao, the test engineer would be well-advised to stop after $\sim 20,000$ tests for peak confidence, rather than run 46,000 for a hypothetical 99% confidence. This would of course be dependent on being able to justify adequate numerical confidence in the test setup, which Section 3 has cast doubt upon.

3.8 Summary of Implementation Challenges

Section 3 has discussed many of the implementation challenges, both technical and programme-based, that face a project looking to use statistical testing to provide quantitative evidence of the reliability of a programmable system. None of the challenges are insurmountable individually, but tackling all of them is a significant endeavour and unlikely to be reasonably practicable to a level of rigour or certainty necessary to support quantification of α . Having to make compromises and judgements to overcome the challenges can lead to tricky qualitative arguments, and risks undermining the targeted quantitative outcome.

IAEA (2018) states that when dealing with high-reliability systems, the dominant uncertainty tends to come from epistemic rather than aleatory uncertainty, due to the high basic assumption of reliability of the system under assessment. This reflects in the described challenges of implementing statistical testing to meet the core conditions — we are limited in the accuracy of the test infrastructure we can produce when we are trying to demonstrate very low failure rates in the test object. Use of CBI (see Sub-section 3.6) illustrates IAEA's point quite clearly.

We conclude that in execution of statistical test campaigns it is easy to underestimate uncertainty, especially in complex systems like a reactor protection system. We see that oversights leading to overconfidence historically seen in other domains (such as in measurement of physical constants, as studied by Henrion and Fischhoff (1986)) may also be pervasive in software engineering.

4 System Complexity and the Safety Case

A picture of the types of systems and components that have been subjected to statistical testing emerges from a literature review. For the most part, the technique has been applied to simpler systems — devices of limited functionality (e.g. Kuball *et alia* (2004), Kuball (2024)), or low-demand/protection functionality subsets of larger programmable systems (e.g. Gough and Kuball (2014), US NRC (2017)). In the latter cases, the testing was limited to the functionality subset due to the challenges associated with statistical testing of high-demand/continuous systems.

When it comes to programmable systems, “complexity” is a disputed term: a system or component with only one primary function may be complex due to its support functions or simply its implementation, for example. Regardless, statistical testing theory is “blind” to the size or complexity of the test object by any metric — a smart device and a reactor protection system will be subject to the same number of tests at a given confidence and reliability target²⁶.

A different perspective on complexity is to consider the size and variety of the operational profile. A simple trip amplifier, which changes its relay output according to a single analogue input’s value compared to a set point (perhaps even including hysteresis), can be tested using a relatively small operational profile — there is very little dimensionality to the input space, and so within the statistical-testing framework there are few unique input scenarios around the point of safety-function demand²⁷. For a reactor protection system, on the other hand, a whole library of distinct plant faults contributes to the operational profile, and each fault may have subtypes depending on the behaviour of dependent and independent plant variables.

Now, we consider what is claimed by the safety case.

It is convenient shorthand to attribute reliability (pfd) figures to a system, but the reality is that safety claims are made on the reliability of *safety functions*. These may be implemented in part or in whole by a system or component, but the reliability that the safety case seeks to demonstrate is the achieved reliability of the safety functions, not the components themselves.

Statistical testing within BS EN 61508-2:2010, for example, is used as a validation activity. Validation is used to show that the system meets “*in all respects the requirements for safety in terms of the required safety functions and safety integrity*”, which means that the safety functions themselves shall be assessed, not the system specifically. Even where statistical testing is used as part of a system integration activity, this exercise is to demonstrate that the subsystems developed to implement the safety function “*interact correctly to perform their intended function*”.

We select or build a system based on the determined reliability requirements of the functions allocated to it. The fault and protection schedule, and associated analysis, will focus on the likelihood of the *function* failing: the system is just part of the implementation of the function. Despite this, statistical testing always appears to be performed as if the system’s reliability is homogeneous, and therefore claims to be a measure of the system reliability, not the *function* reliability²⁸. This system-based approach perhaps originates from descriptions such as Musa (1993), where the objective was to use operational-profile testing to reduce the likelihood of the system encountering faults, rather than demonstrating the reliability of individual functions (that paper discusses how to best use test time to focus on the software functions most likely to be used in real life).

We will now consider a hypothetical system, with safety functionality delivered by three separate internal boards or processors. In our example, each safety function could plausibly

²⁶ Sizewell B PPS contained around 100,000 lines of code (Littlewood et al. 1998), and Hinkley Point C’s PS comprises even more lines (AtkinsRéalis 2020). By comparison, many smart devices (especially those developed for safety industries) are likely to be limited to thousands or perhaps a few tens of thousands of lines of code.

²⁷ Each test case must feature a demand on the safety function(s), but there are only so many distinct ways to approach a setpoint: perhaps by changing ramp rates, or introducing trajectory shapes in the build-up to the demand moment.

²⁸ Several studies suggest that defects are clustered rather than uniformly spread throughout the code. For example, see Hopkins and Hatton (2019), Boehm and Basili (2001) and Daniels and Tudor (2022).

be implemented by its own software on its own processor on its own internal board, all present within a common chassis of a system. This is an extreme version of a common arrangement in a reactor protection system, where the multiple safety functions assigned to the system are distributed over multiple discrete processing units. From the perspective of statistical testing this is a black box and the internal details of implementation do not matter.

We run a statistical testing suite for our hypothetical system, where boards #1 and #2 are tested in nearly every test case due to the frequency of the fault scenarios that demand those functions in the operational profile. Board #3's function is only called a small number of times, if at all, due to the statistical sampling. All tests pass, meaning we claim high confidence in the *system's* reliability. For each safety function, the confidence in function reliability is very different, yet it is this reliability that we actually want to demonstrate. For the safety function implemented by board #3, we have almost zero statistical confidence in its reliability based on how much it was tested. For the safety functions implemented on boards #1 and #2, their confidence will also be slightly lower than the confidence in the overall system. The diminishing confidence per safety function becomes more challenging the more functions that the system implements (and potentially as more categories of scenarios are modelled).

In the fault and protection schedule, the fault's initiating event may have a low frequency, but the reliability of the safety function will always align with the system's overall pfd (e.g. 10^{-3} pfd). For a single-function smart device, there would be no problem, because evidence of the reliability of the safety function and system will be aligned. For a multi-function system like a protection system, to argue that a test suite featuring a handful of tests touching a particular function shows 99% confidence of 10^{-3} pfd *in that function* either: 1) assumes homogeneity of defects within the system (such that each fault scenario has equal chance of revealing a defect); or 2) double-counts the initiating event's own low likelihood. Neither of these arguments are sound: the first is not based on any conventional understanding of software defects, and the second fails to represent the numbers in the fault and protection schedule.

A hypothetical fault and protection schedule for a plant with a protection system with only two safety functions is shown in Table 2. The final column indicates what a $\alpha = 99\%$, $\theta = 10^{-3}$ pfd statistical testing campaign would tell us about the two functions. In reality, some initiating events may demand multiple safety functions, so the functions are tested in multiple scenarios; however most, if not all, functions will not be demanded by every initiating event.

Table 2 ~ Illustrative Fault and Protection Schedule

Initiating event	Initiating-event frequency / y^{-1}	First-line protection	Claimed first-line reliability / pfd	Number of statistical tests	Statistical confidence in first-line function
Fault 1	10^{-1}	Safety function A	10^{-3}	4,560	~99%
Fault 2	10^{-3}	Safety function B	10^{-3}	45	~4%

For context, if using a purely statistical sampling approach, Hinkley Point C PS statistical testing would use ~25,000 tests on the three most common fault scenarios, leaving 25,000 tests for the remaining ~147 scenarios, with the majority of those tests covering only the next 21 most common scenarios (Walker 2024)²⁹. To combat this, the project is reported to be looking to redistribute tests to increase coverage of lower-frequency scenarios without increasing the overall test-suite size (Walker 2024). If the targeted pfd is less onerous (e.g. 10^{-2} pfd) then coverage per function and scenario may be even thinner, reflected in the grading of technique effectiveness by safety integrity level in BS EN 61508-2:2010.

We have not identified any instances where the “enhanced model” binning method proposed by May *et alia* (1995), which is claimed to help address the issue of system complexity and coverage, has been used in a real statistical test campaign. By considering code coverage in the calculations, it makes the statistical-test result more conservative (i.e. more tests required to demonstrate the same pfd), so this will exacerbate the test-suite execution-time challenge. In terms of validity, it still seems to assume homogeneous defect distribution across the system, since the metrics are not sensitive to the aggregated code coverage, and that code coverage is an accurate proxy for state-space coverage. May *et alia* (1995) do not define what code-coverage metric would be used, but this may strongly affect the validity of the code-coverage to state-space-coverage relationship. Further, by binning based on code coverage, this further requires us to move into white-box territory, rather than the black-box territory supposedly targeted by statistical testing. May and Lunn (1995) describe the enhanced model in more detail, and acknowledge that more work is needed to account for code complexity (and hence distribution of sources of failure) in the enhanced model.

Finally, regarding the use of statistical testing as evidence to support the safety case, it is not clear how to reconcile the statistical testing approach with the fact that these systems can also be interacted with by reactor operators, a challenge discussed by Strigini and Littlewood (1997). The technique does not have a developed means of realistically modelling human interactions with the system³⁰, and thus any human-interface functionality is not considered as part of the modelled operational profile (socio-technical aspects are ignored). To some extent this is reasonable, because in a reactor fault the automatic protection functions are expected to negate the need for operator interaction; however, this assumes that the system is in a predictable state unfettered by human interaction at the start of and during a transient. For systems with demands where human interaction is likely, this becomes more significant. Any human interaction, whether concurrent or in advance, may ultimately affect the internal state of the system. To exclude human interactions in the estimation of reliability must implicitly assume that the system development lifecycle is equally good at implementing and verifying manual functions as it is at implementing and verifying automatic functions.

We conclude that it is not clear how to measure the value of the quantitative reliability estimate delivered by statistical testing when building a safety case for complex systems, and nor is it clear how to align the statistical findings with what the safety functions actually need. We conclude that to actually support the safety case’s claims on the safety functions, it would be necessary to run a test suite *for each safety function*, each of the size given by Table 1. For a realistic protection system, this would likely be intractable due to the number

²⁹ Note that the material in Walker (2024) was presented at a publicly accessible IAEA conference, but is not in a readily accessible or peer-reviewed form. For this reason, these values should be taken as indicative/illustrative and should not be quoted without that qualification. Based on our experience of similar systems, we posit that these figures would follow similar patterns for the majority of statistical-testing campaigns.

³⁰ For example, Gough and Kuball (2014) describe how simplifications were needed to handle operator resetting alarms, which was acknowledged to not be fully realistic. That this relatively simple interaction was challenging highlights why a human-interaction model is often omitted.

of safety functions typically allocated to these system — both Sizewell B PPS and Hinkley Point C PS literature indicates that time-to-complete is a significant consideration, so even a doubling of test time would be unachievable³¹.

5 What Can We Learn from the Statistical Testing Method?

Despite the concerns outlined so far in this paper regarding quantitative evidence, statistical testing practices can be useful in developing qualitative evidence of system quality. This seems inarguable, as statistical testing usually involves a campaign of tests that are more holistic than testing performed during the system development lifecycle.

Kuball *et alia* (2004) describe how their test campaign using statistical testing techniques found a defect that several other test techniques did not. This is positive, but we argue that this is not because of the mathematical statistical testing theory but rather due to the method of sampling, which was not a pure operational profile³². Lessons learned from how their test campaign was structured from a functional perspective (i.e. upon what types of scenarios they focussed) may be used to help build future campaigns based on operational profiles, even if the statistical testing theory, e.g. Equation (1), is not used to unilaterally dictate the size or distribution of the campaign. The work of Kuball *et alia* focussed tests on combinations of error conditions, rather than what statistical testing would usually do, which is to sample tests based purely on their frequency in the operational profile.

A different technique that may achieve similar results to Kuball *et alia* (2004) is combinatorial (t-way) testing (Kuhn 2011). Like the test campaign of Kuball *et alia*, it aims to determine combinations of conditions that may reveal defects. Techniques have been developed in the field of combinatorial testing to achieve an efficiently sized test suite that minimises over-coverage of combinations of conditions. Note that combinatorial testing, while providing a rigorous approach, does not provide a quantitative reliability measure.

As described by Kuball (2024), a statistical testing campaign for a spine-break detection system (claimed at 10^{-2} pfd) revealed discrepancies between the system's documentation and its behaviour with respect to its previously less-tested rate-of-change alarm. This demonstrated the benefit of detailed testing of a device's individual functions, and of not assuming that successful testing of function A implies equal quality in function B.

Kuball and May (2007) describe work in a statistical testing project to identify how to initialise the test object between tests to ensure non-interference between tests. This involved review of device documentation, and a series of tests to build an understanding of the length of the internal memory's history. This was a black-box approach, *inferring* the history length. For simpler devices, this inference may have a reasonable degree of confidence, but for larger systems the ease of building confidence will diminish. Again, the

³¹ While the actual increase in test time is inherently and strongly dependent on application context, we argue a doubling of suite size would be optimistic due to relation of functions (for example, a function protecting against high pressure in a vessel is unlikely to be demanded in the same test as a function protecting against low pressure in that vessel). That Tooley and Stokes (1992) reported that their set of scenarios only tested “a majority” of safety functions indicates that some functions are demanded in only specific circumstances.

³² Their test suite was more challenging than a randomly sampled statistical testing suite because test cases were excluded if they had already been covered by earlier test phases. This was pragmatic, and meant that the statistical test suite could focus on the more onerous scenarios rather than just test the most common/less challenging scenarios. Exclusion of test cases from the statistical testing profile would require analysis to demonstrate exhaustiveness of existing testing.

approach taken is not something that is exclusive to statistical testing, but it is a technique that could be useful for other forms of testing.

We believe that, given the difficulties in providing enough quantitative confidence that assumptions are fulfilled and implementing the statistical-testing campaign to achieve this, the target for programmable-systems assurance should not be to determine a quantitative measure of reliability, but to build a qualitative confidence in the system and its implementation of its safety functionality. By avoiding a focus on the quantitative objective, this may open the door to more valuable testing that generates greater confidence-building value, perhaps even with less cost and effort (for example by altering the distribution and content of tests, or allowing simulation testing of emulated objects to contribute to the outcome).

6 Conclusion

We have argued that in the case of complex systems, such as reactor protection systems, it is not reasonably practicable to adequately achieve all the statistical testing core conditions to a sufficient level, so that the confidence value derived from the results would have supportable quantitative meaning³³. Past attempts at using complex statistics to demonstrate that qualitative arguments do not invalidate the quantitative claims tend to be limited to hard-to-scale model cases and, even then, these attempts often rely on qualitative assumptions.

Our position does not appear to be unique, being reflected for example in the (non-system-specific) notes in nuclear C&I systems standard BS EN 61513 that “*there are no generally recognised means to quantify the reliability of software*” and in generic functional-safety standard BS EN 61508-3:2010 that “*the probability of occurrence of systematic failures cannot in general be quantified*”. A literature survey produced for the ONR (ATLAS Consortium 2023) assessed that “*no single method alone is considered sufficient to accurately estimate software reliability*”.

We conclude that, ignoring challenges around test-case independence, the limit of what we can quantitatively claim from a traditionally executed statistical testing campaign of size N is the following:

The probability θ , to a statistical confidence α , that a system (identical to the test object in version, configuration and state of condition) with one safety function (or multiple decoupled safety functions that are demanded in every test) will, when subjected to test-harness stimulation (selected from the test distribution derived from the model of the real operational profile), respond in a manner that aligns with the predictions of the oracle’s model of the system requirements.

If statistical testing were performed in a closed loop (i.e. the outputs of the test object can influence subsequent simulation steps) and the pass/fail determination were based on whether the simulated plant is safe, then the final clause would be:

... will respond in a manner that results in the simulated plant reaching a safe state.

³³ A judgement specifically on the appropriateness of statistical testing for digital devices of limited functionality (as defined in BS IEC 62671:2013 and commonly referred to as smart devices) is beyond the scope of this paper.

Note that both open- and closed-loop testing conclusions above are still limited to the test environment — information about how they behave in the real environment can only be inferred based on understanding of the closeness of the two environments.

Inadequacies in the fidelity of the operational profile, plant simulation, oracle, test harness, or representativeness of the test object are either unquantifiable or can only be approximated, so would only serve to reduce the confidence α by an unknowable amount when trying to translate this outcome to the real system. This is further complicated by considerations of system and plant ageing, maintenance (e.g. see Ladkin and Littlewood (2016)), modifications, and the effect of human interaction with the system and plant. As seen in the CBI approach in Littlewood and Wright (2007) and Salako and Zhao (2023), the confidence reduction associated with inaccurate assumptions can be stark and hard to quantitatively predict. In large multi-function systems, are we just over-fitting to an under-scoped inaccurate test operational profile, and in single-function devices are we just assuming that by running thousands of tests we would more likely find a defect in the device’s safety function than a device defect being masked by an inadequate test oracle³⁴?

We question why the target confidence α is seemingly always 95% or 99%³⁵, when, for UK nuclear at least, the reliability claim placed on the system is a 50% confidence “best-estimate” in the probabilistic safety analysis. The additional tests performed to take the statistical testing from 50% to 95% (or 99%) confidence provide diminishing levels of real value, and the dependence on the core conditions becomes increasingly strong. For a 10^{-4} pfd system, Table 1 showed that the difference between 50% and 99% confidence is 39,000 tests; however, many of these extra tests will be of the same few scenarios that would have already been tested quite extensively within the limits of the test operational profile’s fidelity³⁶.

We also conclude that statistical testing is focussed on providing a measure of the implemented *system* reliability, yet the safety case (and indeed BS EN 61508) is primarily concerned with the reliability of each *safety function*. This distinction has not previously been discussed in statistical-testing literature to the best of our knowledge. Statistical testing therefore relies on flawed deductive reasoning to support the safety case, such as:

Premise 1 Statistical testing provides confidence α that the system reliability is at least as good as θ when subjected to its operational profile.

Premise 2 The system implements one or more safety functions.

Conclusion We have confidence α that each safety function implemented by the system has reliability at least as good as θ when subjected to the system’s operational profile.

The first premise means that there the probability that the “next” simulated plant fault will reveal a system failure does not exceed θ . We consider Fault X: our operational profile tells us that Fault X will happen at a frequency of 10^{-x} y^{-1} . The safety case is interested in whether the protection system will act correctly when Fault X happens. But, if Fault X is infrequent, then its associated safety functions have received minimal testing, so we have minimal

³⁴ This sentence swings both ways — the two questions could apply to both types of test object.

³⁵ Indeed, NS-TAST-GD-046 recommends this target.

³⁶ Taking the illustrative numbers in Walker (2024), a 50% confidence suite might test the most common scenario over 2,000 times. Do we get more confidence by increasing the suite to 99% confidence, which will include running another ~13,000 tests of that scenario? A single omission from the operational profile’s description of that scenario, e.g. a human interaction with the system, or an unmodelled fault, or misconfiguration, could completely change the system behaviour. Further, as noted by Salako and Zhao (2023), more tests may reduce confidence if there is any doubt in the test setup.

confidence on that safety function's reliability. The above premises can only be used to reach the conclusion if we assume that software-based systems only fail randomly, not systematically, or that the probability of a rarely used function failing *on demand* is the same as a frequently used function³⁷. Otherwise, we suggest that the statistical-testing suite needs to be big enough that each safety function is demanded by N tests, rather than N tests for the system as a whole.

Musa (1993) provides figures extolling the commercial benefit of operational-profile testing in commercial software development, including as part of the development lifecycle. The numbers are compelling, yet it is hard to find similar evidence for statistical testing of *high-integrity* programmable systems, especially when they have already completed validation. From all of the literature studied, Kuball *et alia* (2004) declare three (non-safety-related) defects identified by statistical testing for a safety interlock system, and other studies noted by Kuball (2024) refer to discrepancies from user documentation, or lack of detail in documentation, leading to configuration oversights detected by the test campaigns in commercial off-the-shelf products. While Kuball *et alia* (2004) argue the secondary benefit of statistical testing finding the other defects more efficiently than the other techniques in their study, it is not clear that statistical testing would have been *the* most efficient way to achieve the result. For example, combinatorial testing covering arrays can be calculated to find the most efficient way of covering variable interactions and, alternatively, more efficient defect-finding might be achieved by the “*enhanced functional testing*” described in Kuball (2024), akin to “*expanded functional testing*” as described in BS EN 61508-7:2010.

We conclude that in practice, statistical testing should not be pursued as a technique for *quantitatively* demonstrating system reliability. If we remove the objective of a quantitative result, we lose several constraints, and so open the door for extracting more value out of testing, such as use of simulators/emulators to increase test throughput, use of instrumentation to measure coverage, and more efficient exploration of state space. Walker (2024) indicates that such movement towards qualitative test benefit is already being considered in large real-life projects.

We argue that this proposition for forgoing statistical testing does not necessarily require a direct replacement for the technique (i.e. another approach to quantitatively demonstrating reliability of a software-based system), because our assertion is that, for the majority of systems (especially many-function systems), quantitative demonstration of reliability is not in fact achieved with statistical testing today anyway, for the reasons outlined in this paper.

All is not lost, however, because without the constraints required for quantitative analysis, there may time and resources available to extract a benefit from performing other techniques such as operational-profile testing prior to C&I system validation to demonstrate behavioural properties. This would be for the purposes of finding and fixing defects rather than developing numerical perceived confidence in the numerical confidence in reliability. Such pragmatic techniques can serve to actually increase the quality of the system and work on minimising systematic defects, and then eventually to provide qualitative confidence that operational use will not reveal further defects, building on the work of Musa (1993).

³⁷ We argue that it is not valid to assume that all functions are equally defective (or undefective) due to the concept of defect clustering.

7 Further Work

Some of the principles of test infrastructure and test suite design developed to support statistical testing over the years are useful, and there is some evidence to suggest they have benefits over other forms of testing. However, it is proposed that they should be used with the objective of targeting valuable and justifiable qualitative evidence of the reliability of safety functions, complementing other forms of quantitative evidence of the functional correctness of the system and its functions.

As well as considering the value that pre-validation operational-profile testing of software may offer, research into other techniques such as combinatorial testing (Kuhn 2011) (Diemert et al. 2023) or expanded functional testing (BS EN 61508 7:2010) may also supplement this work and inform testing approach.

This paper has focussed on larger, complex systems like reactor protection systems. Smaller, simpler systems or components, such as digital devices of limited functionality (smart devices), may not be as sensitive to some of the arguments made in this paper, and so may benefit more from a “purer” statistical-testing approach. Further work is needed to determine whether statistical testing has value for such devices, or whether operational-profile testing, combinatorial testing, or another option are more appropriate. It is noted that overall safety arguments for and against statistical testing of smart devices may have different structural considerations compared to larger systems and a holistic approach to the smart devices’ safety justification would be needed. We ask what “reliability” for a smart device when integrated into a system means. How does one determine the pfd required for a smart device within a system — if the safety function reliability is claimed as 10^{-2} pfd, then what would be the smart device’s “reliability” in its integrated environment — presumably better than 10^{-2} pfd unless the rest of the system is flawless? What benefit is there of determining the “reliability” of a smart device, if the safety function to which it contributes also requires the action of the wider system in which the smart device is integrated? Failure of the smart device to respond to a demand may not prevent the safety function being performed, and conversely, the safety function may still not be achieved by the system even if the smart device behaved according to its specification.

Disclaimers

Opinions expressed herein are those of the authors, not necessarily those of their past and present employers or clients. The authors note, however, that our past and present employers and clients may materially benefit if the conclusions of this paper are more widely accepted.

Acknowledgments

The authors would like to thank Dan Finnigan (Rolls-Royce SMR), Lee Walker (Rolls-Royce SMR), and Waleed Chaudhry (Holtec Britain) for their comments on an early draft of this article.

The authors would also like to express their gratitude to the anonymous peer reviewers who provided supportive, thoughtful, and constructive commentary on this article, improving its robustness, accuracy and value.

References

- Arthur C. (1993). *Extra tests forced on Sizewell's safety software*. New Scientist, 6th November 1993. Available from <https://www.newscientist.com/article/mg14018980-600-extra-tests-forced-on-sizewells-safety-software/>. Accessed 20th December 2025.
- ASCA. (2007). *Risk-informed safety assurance and probabilistic risk assessment of mission-critical software-intensive systems*. ASCA Inc., Report AR 07-01. Available from <https://ntrs.nasa.gov/api/citations/20100006923/downloads/20100006923.pdf>. Accessed 16th September 2026.
- AtkinsRéalis. (2020). *Atkins to verify the safety of reactor protection software at Hinkley Point C*. Trade Release of 23rd April 2020. <https://www.atkinsrealis.com/en/media/trade-releases/2020/2020-04-23>. Accessed 12th February 2026.
- ATLAS Consortium. (2023). *Provision of technical support to the consideration of realistic software modelling in PSA*. ATLAS Consortium, 3rd March 2023. Available from <https://www.onr.org.uk/publications/regulatory-reports/research/research-reports/onr-rrr-112>. Accessed 13th February 2026.
- Bishop P, and Povyakalo A. (2017). *Deriving a frequentist conservative confidence bound for probability of failure per demand for systems with different operational and test profiles*. Reliability Engineering & System Safety 158:246-253.
- Boehm B, and Basili V. R. (2001). *Software Defect Reduction Top 10 List*. IEEE Computer, 34(1):135-137.
- BS EN 60880. (2009). *Nuclear power plants – Instrumentation and control systems important to safety – Software aspects for computer-based systems performing category A functions*. BS EN 60880:2009. British Standards Institution, London.
- BS EN 61508-2. (2010). *Functional safety of electrical/electronic/programmable electronic safety-related systems. Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems*. BS EN 61508-2:2010. British Standards Institution, London.
- BS EN 61508-3. (2010). *Functional safety of electrical/electronic/programmable electronic safety-related systems Part 3: Software requirements*. BS EN 61508-3:2010. British Standards Institution, London.
- BS EN 61508-4. (2010). *Functional safety of electrical/electronic/programmable electronic safety related systems. Part 4: Definitions and abbreviations*. BS EN 61508-4:2010. British Standards Institution, London.
- BS EN 61508-7. (2010). *Functional safety of electrical/electronic/programmable electronic safety-related systems Part 7: Overview of techniques and measures*. BS EN 61508-7:2010. British Standards Institution, London.
- BS EN 61513. (2013). *Nuclear power plants – Instrumentation and control important to safety – General requirements for systems*. BS EN 61513:2013. British Standards Institution, London.
- BS IEC 62671. (2013). *Nuclear power plants – Instrumentation and control important to safety – Selection and use of industrial digital devices of limited functionality*. BS IEC 62671:2013. British Standards Institution, London.

- Butler R. W, and Finelli G. B. (1993). *The infeasibility of quantifying the reliability of life-critical real-time software*. IEEE Transactions on Software Engineering 19(1):3-12.
- Coolen F. (2012). *On some statistical aspects of software testing and reliability*. Advances in Intelligent and Soft Computing 170:103-113.
- Daniels D, and Tudor N. (2022). *Software Reliability and the Misuse of Statistics*. Safety-Critical Systems eJournal. Vol.1, Iss. 1. Available from <https://scsc.uk/r1518>. Accessed 16th September 2026.
- Diemert S, Casey A, and Robertson J. (2023). *Challenging Autonomy with Combinatorial Testing*. In: *2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pp. 214-222. Dublin, Ireland.
- Dolan B. (1994). *Software testing at Sizewell*. The RISKS Digest, Volume 15 Issue 58, 23rd February 1994. Available from <https://catless.ncl.ac.uk/Risks/15/58#subj5.1>. Accessed 20th December 2025.
- EN 50129. (2003). *Railway applications — Communication, signalling and processing systems — Safety related electronic systems for signalling*. EN 50129:2003, CENELEC (European Committee for Electrotechnical Standardization), Brussels.
- EN 50716. (2023). *Railway Applications — Requirements for software development*. EN 50716, 1st Edition, 2023. CENELEC (European Committee for Electrotechnical Standardization), Brussels.
- EPRI. (2008). *Operating experience insights on common-cause failures in digital instrumentation and control systems*. Electric Power Research Institute, Inc. Available from <https://www.epri.com/research/products/1016731>. Accessed 1st July 2026.
- EPRI. (2010). *Estimating failure rates in highly reliable digital systems*. Electric Power Research Institute, Inc. Available from <https://www.epri.com/research/products/0000000000001021077>. Accessed 1st July 2026.
- Gough H, and Kuball S. (2014). *Application of Statistical Testing to the Data Processing and Control System for the Dungeness B Nuclear Power Plant (Practical Experience Report)*. In *Proceedings of the 10th European Dependable Computing Conference, EDCC 2014*. pp158-165.
- Hansard. (1993). *House of Commons debate*. In: Commons Chamber Volume 233. November to December 1993.
- Henrion M, and Fischhoff B. (1986). *Assessing Uncertainty in Physical Constants*. American Journal of Physics 54(9):791-798.
- Hopkins T, and Hatton L. (2019). *Defect patterns and software metric correlations in a mature ubiquitous system*. arXiv 6th December 2019. Available from <https://arxiv.org/abs/1912.04014>. Accessed 16th February 2026.
- Horgan J, and Mathur A. (1996). *Chapter 13: Software Testing and Reliability*. In: Lyu M. R. (Editor), *Handbook of Software Reliability Engineering*. IEEE Computer Society Press and McGraw-Hill Book Company, New York, USA. pp. 531-566.

- Hunns D. M, and Wainwright N. (1992). *Software-based protection for Sizewell B: The regulator's perspective*. In: International Conference on Electrical and Control Aspects of the Sizewell B PWR, (IEE Number no.361), Institution of Electrical Engineers, British Nuclear Energy Society, Institute of Energy, Institution of Mechanical Engineers, and Convention of National Societies of Electrical Engineers of Western Europe (EUREL), London.
- IAEA. (1999). *Verification and validation of software related to nuclear power plant instrumentation and control*. International Atomic Energy Agency, Technical Reports Series No. 384. <https://www.iaea.org/publications/5718/verification-and-validation-of-software-related-to-nuclear-power-plant-instrumentation-and-control>. Accessed 16th February 2026.
- IAEA. (2018). *Dependability assessment of software for safety instrumentation and control systems at nuclear power plants*. International Atomic Energy Agency, Nuclear Energy Series No. NP-T-3.27. <https://www.iaea.org/publications/12232/dependability-assessment-of-software-for-safety-instrumentation-and-control-systems-at-nuclear-power-plants>. Accessed 11th February 2026.
- Ichien N. M. (1998). *Software in Safety Applications*. International Symposium on Evolutionary Water Cooled Reactors: Strategic Issues, Technologies, and Economic Viability. Available from <https://inis.iaea.org/records/3tfe8-def24/preview/30020332.pdf>. Accessed 11th February 2026.
- IEC61508. (2010). *Functional safety of electrical/electronic/programmable electronic safety-related systems*. IEC 61508, in 7 parts, 2nd Edition, 2010. International Electrotechnical Commission, Geneva.
- IEEE 352. (2016).: *IEEE Guide for General Principles of Reliability Analysis of Nuclear Power Generating Station Systems and Other Nuclear Facilities*. IEEE 352-2016. Institute of Electrical and Electronics Engineers, New York.
- IEEE 1633. (2016). *IEEE Recommended Practice on Software Reliability*. IEEE 1633-2016. Institute of Electrical and Electronics Engineers, New York.
- IRSN. (2017). *Principles relating to the digital instrumentation and control design approach*. Institut de Radioprotection et de Sûreté Nucléaire. Available from https://research-assessment.asnr.fr/sites/en/files/2023-12/IRSN_Safety_approaches_digital-instrumentation-and-control_2017.pdf. Accessed 1st July 2026.
- ISO 26262. (2018). *Road vehicles — Functional safety*. ISO 26262:2018, in 12 parts, 2nd Edition, 2018. International Organization for Standards, Geneva.
- Knight J, and Leveson N. (1986). *An experimental evaluation of the assumption of independence in multiversion programming*. IEEE Transactions on Software Engineering SE-12(1).
- Korhonen J, Pulkkinen U, and Haapanen P. (1997). *Statistical reliability assessment of software-based systems*. Finnish Centre for Radiation and Nuclear Safety, Report number STUK-YTO-TR--119.
- Kuball S. (2024). *Black box testing for functional safety*. T6A Symposium 2024 — Guidance in Functional Safety Compliance. The 61508 Association. Available from <https://61508.org/wp-content/uploads/2024/11/08A-T6A-Event-2024-A08-SKuball.pdf>. Accessed 16th September 2026.

- Kuball S, Hughes G, May J, Gallardo J, and John A. (2004). *The effectiveness of statistical testing when applied to logic systems*. Safety Science 42:369-383.
- Kuball S, and May J. (2007). *A discussion of statistical testing on a safety-related application*. In: Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability. 2007;221(2):121-132.
- Kuhn R. (2011). *Introduction to combinatorial testing*. National Institute of Standards and Technology, Presentation at Carnegie-Mellon University, 7th June 2011. Available from <https://mse.s3d.cmu.edu/facstaff/faculty1/faculty-publications/miranda/kuhnintroductioncombinatorialtesting.pdf>. Accessed 18th December 2025.
- Ladkin P. B, and Littlewood B. (2016). *Practical statistical evaluation of critical software*. Paper presented at the 24th Safety-Critical Systems Symposium, Brighton, UK. February 2016. Available from <https://rvsbi.de/publications/Papers/LadkinLittlewood-Practical-FinalV2.pdf>. Accessed 16th February 2026.
- Leveson N. (2004). *A new accident model for engineering safer systems*. Safety Science 42:237-270.
- Littlewood B, Bainbridge I, and Bloomfield R. (1998). *The use of computers in safety-critical applications*. Health and Safety Commission, London, UK.
- Littlewood B, and Strigini L. (1993). *Validation of ultrahigh dependability for software-based systems*. Communications of the ACM 36(11):69-80. Available from <https://openaccess.city.ac.uk/id/eprint/1251/1/CACMnov93.pdf>. Accessed 16th September 2026.
- Littlewood B, and Wright D. (1995). *Stopping Rules for the Operational Testing of Safety-Critical Software*. Digest of Papers, Twenty-Fifth International Symposium on Fault-Tolerant Computing. Pasadena, USA.
- Littlewood B, and Wright D. (1997). *Some conservative stopping rules for the operational testing of safety-critical software*. IEEE Transactions on Software Engineering 23(11):673-683.
- Littlewood B, and Wright D. (2007). *The use of multilegged arguments to increase confidence in safety claims for software-based systems: A study based on a BBN analysis of an idealized example*. IEEE Transactions on Software Engineering 33(5):347-365.
- Lyu M. R. (Ed.) (1996). *Handbook of software reliability engineering*. IEEE Computer Society Press and McGraw-Hill Book Company, New York, USA . Available from <https://www.cse.cuhk.edu.hk/~lyu/book/reliability/>. Accessed 1st July 2026.
- May J. H. R. (2017). *Theoretical feasibility of statistical assurance of programmable systems based on simulation tests*. In: 2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Prague, Czech Republic.
- May J. H. R, Hughes G, and Lunn A. D. (1995). *Reliability estimation from appropriate testing of plant protection software*. Software Engineering Journal, November 1995:206-218.
- May J. H. R, and Lunn A. D. (1995). *A model of code sharing for estimating software failure on demand probabilities*. IEEE Transactions on Software Engineering 21(9):747-753.

- Miranda B, and Bertolino A. (2017). *An assessment of operational coverage as both an adequacy and a selection criterion for operational profile based testing*. Software Quality Journal 26:1571-1594.
- Musa J. (1993). *Operational profiles in software-reliability engineering*. IEEE Software 10(2):14-32.
- ONR. (2011a). *Generic Design Assessment - New Reactor Build: Step 4 Control and Instrumentation Assessment of the Westinghouse AP1000® Reactor*. Office for Nuclear Regulation, Assessment Report ONR-GDA-AR-11-006.
- ONR. (2011b). *UK EPR GDA Project – Resolution Plan for GI-UKEPR-CI-02*. Office for Nuclear Regulation, Resolution Plan GI-UKEPR-CI-02-RP. Available from <https://www.onr.org.uk/media/2mhmjqly/resolution-plan-gi-ukepr-ci-02.pdf>. Accessed 16th September 2026.
- ONR. (2012). *New nuclear reactors: Generic Design Assessment - Electricité de France SA and AREVA NP SAS UK EPR™ nuclear reactor - Summary of the GDA Issue close-out assessment of the Electricité de France SA and AREVA NP SAS UK EPR™ nuclear reactor*. Office for Nuclear Regulation, Assessment Report ONR-GDA-SR-12-001.
- ONR. (2017). *New Reactor Division: Step 4 Assessment of Control and Instrumentation for the UK Advanced Boiling Water Reactor*. Office for Nuclear Regulation, Assessment Report ONR-NR-AR-17-017.
- ONR. (2020). *Independent Confidence Building Measures for complex control and instrumentation systems*. Office for Nuclear Regulation, Regulatory Observation RO-UKHPR1000-0057.
- ONR. (2023). *Computer Based Safety Systems*. Office for Nuclear Regulation. Technical Assessment Guide NS-TAST-GD-046. Available from <https://www.onr.org.uk/media/3def2aat/ns-tast-gd-046.docx>. Accessed 16th September 2026.
- ONR. (2024). *Licensing of safety critical software for nuclear reactors – common position of international nuclear regulators and authorised technical support organisations*. Regulator Task Force on Safety Critical Software (TF SCS) Report. Available from <https://www.onr.org.uk/publications/regulatory-reports/other-reports/licensing-of-safety-critical-software-for-nuclear-reactors>. Accessed 25th February 2026.
- Ontario Power Generation (2013). *Standard for Software Engineering of Safety Critical Software*. Ontario Power Generation Inc., CANDU (Computer Systems Engineering Centre of Excellence), CE-1001-STD Revision 3.
- Prokop L. (2024). *Historical aerospace software errors categorized to influence fault tolerance*. In: IEEE Aerospace Conference, Big Sky, USA.
- RTCA/DO-178C. (2012). *Software Considerations in Airborne Systems and Equipment Certification*. RTCA Inc., Washington DC.
- SAE ARP4754A. (1996). *Recommended Practice, Guidelines for Development of Civil Aircraft and Systems*. ARP4754A. SAE International, Warrendale, PA. Available at <https://doi.org/10.4271/ARP4754A>. Accessed 13th July 2026.
- Salako K, and Zhao X. (2023). *The Unnecessity of Assuming Statistically Independent Tests in Bayesian Software Reliability Assessments*. IEEE Transactions on Software Engineering 49(4):2829-2838.

- Strigini L, and Littlewood B. (1997). *Guidelines for Statistical Testing*. ESA/ESTEC, PASCON/WO6-CCN2/TN12. Available from <https://openaccess.city.ac.uk/id/eprint/254/>. Accessed 1st July 2026.
- Tooley P. A, and Stokes P. (1992). *Dynamic testing of the Primary Protection System*. In: International Conference on Electrical and Control Aspects of the Sizewell B PWR, (IEE Number no.361), Institution of Electrical Engineers, British Nuclear Energy Society, Institute of Energy, Institution of Mechanical Engineers, and Convention of National Societies of Electrical Engineers of Western Europe (EUREL), London.
- UK HSE. (2003). *Out of control: Why control systems go wrong and how to prevent failure*. United Kingdom Health & Safety Executive. Health & Safety Guidance, HSG238. Available from <https://www.hse.gov.uk/pubns/books/hsg238.htm>. Accessed 1st July 2026.
- US NRC. (2017). *Development of a statistical testing approach for quantifying safety-related digital system on demand failure probability*. United States Nuclear Regulatory Commission. Available from <https://www.nrc.gov/docs/ML1715/ML17151A215.pdf>. Accessed 1st July 2026.
- US NRC. (2022). *US Nuclear Regulatory Commission's approach to software reliability including consideration of common-cause failures*. Technical Meeting on the Software Reliability of Digital Instrumentation and Control Systems for Nuclear Power Plant Safety, 13th—16th December 2022. International Atomic Energy Agency, Vienna.
- US NRC. (2026). *(Availability of) An International Report on Safety Critical Software for Nuclear Reactors by the Regulator Task Force on Safety Critical Software (TF-SCS)*. United States Nuclear Regulatory Commission. International Agreement Report NUREG/IA-0463. Available from https://www.govinfo.gov/app/details/GOVPUB-Y3_N88-PURL-gpo67557. Accessed 12th August 2026.
- Walker L. (2024). *Case studies in operational profile based statistical testing for the quantitative reliability estimation of software based commercial off the shelf (COTS) equipment*. Technical Meeting on Commercial Equipment in Nuclear Applications and Non-Nuclear Standards for Safety Instrumentation and Control Systems, 1st—3rd July 2024. International Atomic Energy Agency, Vienna.