SANDIA REPORT
SAND2002-1306
Unlimited Release
Printed May 2002

Mathematical Aspects of Unique Signal Assessment

J. Arlin Cooper

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/ordering.htm

SAND2002-1306
Unlimited Release
Printed May 2002

Mathematical Aspects of Unique Signal Assessment

J. Arlin Cooper[1]
Airworthiness Assurance Department
Sandia National Laboratories
P.O. Box 8500
Albuquerque, NM 87185-0490

Key words: unique signal, ENDS, safety analysis

Abstract

This report is a supplement to "The Unique Signal Concept for Detonation Safety in Nuclear Weapons," SAND91-1269, which provides a prerequisite fundamental background on the unique signal (UQS) concept. The UQS is one of the key constituents of Enhanced Nuclear Detonation Safety (ENDS), as outlined in Section 1 of that report. There have been many documents written over the past quarter of a century describing various aspects of the UQS, but none of these emphasized the mathematical approaches that help explain why the UQS is effective in resisting inadvertent pre-arming, even in abnormal environments and how UQS implementations can be quantitatively assessed. The intent of this report is to describe various pertinent mathematical methodologies (many of which have not been previously reported) without duplicating, any more than necessary, background information available in other reports. Mathematical UQS analysis is needed because of quantitative requirements associated with ENDS, and because limited comparisons of various implementation approaches can be quantified under mathematical modeling assumptions.

Some of the mathematics-based results shown in this report are presented to explain:

1. The reasons that the UQS methodology can provide greater protection against accident environments than could combinational techniques (Sections 2.1 through 2.4)
2. The reason that the probability of inadvertently duplicating a UQS comprising n bi-valued events cannot be estimated as low as $\left(\frac{1}{2}\right)^n$ (Section 2.4)
3. The value of, and the Sandia National Laboratories policy on independent sequential communication of UQS events (Section 3.4)
4. The care that must be exercised if any signal processing is necessary (Section 4).

There are also numerous examples (*e.g.*, in Appendices A and B) of ill-advised deviations from UQS methodology that can seriously degrade safety. These examples help demonstrate that the UQS methodology should not be compromised.

[1] An author biography is given on pg. 68.

Acknowledgment

A large number of mathematically oriented exchanges assisted in the approximately 25 years of developments described in this report. The contributors included Curtis Mueller, Stan Spray, Bill Stevens, Gus Simmons, Jay Grear, Garth Maxam, and Chuck Williams (retired Sandians); former Sandians Wally Crammond and Tom Edrington (both deceased); former Sandian Jackie Leyland (now with Cato Institute); Doug Cooper (Oresis Communications); Scott Ferson (Applied Biomathematics); and Sandians Ken Chen (12332), Perry D'Antonio (9713), John Covan (6252), Kathleen Diegert (12335), Floyd Spencer (12323), Alice Johnson (8241), Don Schroeder (9411), Michele Caldwell (2113), and Jerry Cashen (8418). Stan, Curt, Wally, and Jay were the main UQS mathematics methodology visionaries. Todd Jones (12333) arranged for the support required to produce this report.

Contents

List of Figures

List of Tables

1. **Background**

The function of the Unique Signal (referred to as the UQS in this report[2]) is to provide an extremely high level of resistance to inadvertent pre-arming, even in abnormal environments[3], while reliably providing an enabling stimulus (pre-arm) to a nuclear weapon for normal-environment use. The details of and reasons for the UQS methodology are given in Ref. 1, which is a prerequisite for the material in this report.

1.1. Requirements

The modern quantitative parameters for the UQS (24 bi-valued "events") had their genesis in the jointly (DoD, DOE, and Sandia) agreed-on abnormal-environment requirement that is part of the "Walske letter"[4] of 1968. The abnormal-environment requirement[5] is that "The probability of a premature nuclear detonation … shall not exceed 1 in 10^6 per … exposure or accident." This requirement places a very high demand on the Sandia National Laboratories weapons systems, which must respond safely, even given an exposure or accident. "Abnormal environments" were at that time considered to be " … those environments as defined in the weapon's stockpile-to-target sequence and military characteristics in which the weapon is not expected to retain full operational reliability." The modern understanding is " … vanishingly small risk of a nuclear detonation *given* exposure to any credible abnormal environment," with a 10^{-6} "threshold of acceptability". Since "credible" and "vanishingly small" are imprecise terms, they cannot be precisely mathematical modeled. But UQS response to a large variety of potential threats can be treated mathematically by assuming threat models.

Sandia systems personnel (in consultation with safety personnel) decided that the 10^{-6} per exposure level was unattainable with a single safety device, requiring two abnormal-environment safety subsystems in the ENDS (Enhanced Nuclear Detonation Safety) approach, with the aim of making each significantly better than 10^{-3} per exposure, and engineering a high degree of independence[6] between the two subsystems. This meant that there would be a separate[7] UQS for each abnormal-environment safety subsystem. Each UQS was to be applied to its own "stronglink switch." The safety responsibility for each of the two abnormal-environment safety subsystems is on the information incompatibility of each UQS and the more difficult isolation/inoperability protection of the exclusion regions and stronglink switches (as discussed in Section 2.4).

The basic details of UQS methodology were determined in the early 1970s. By the early 1980s the goal had become to have two human-initiated UQS event sequences (each having a different and unrelated pattern of events), one for each abnormal-environment safety subsystem (double intent). The reasons for unrelated patterns are discussed in

---

[2] A Glossary is on pg. 69.

[3] Abnormal environments transcend normal operating environments, including all degrees of severity.

[4] Carl Walske was then the DoD Military Liaison Chairman and Assistant to the Secretary of Energy.

[5] There is also a normal-environment requirement (10^{-9} over weapon lifetime).

[6] The relation $10^{-3} \times 10^{-3} = 10^{-6}$ is not defensible unless the two subsystems are independent.

[7] Also to be read as "unrelated." This is addressed in Section 2.6.

Section 2.6, and the reasons for double intent are discussed in Appendix A3. A form of double intent was first implemented in the Pershing II[8]. Preparations for double intent were also made in the Sea Lance, SRAM II, and SRAM T programs; and the W89 and W91 would have been double-intent-capable from the warhead interface.

Each modern UQS has 24 separately communicated bi-valued events, in order to be compatible with the extremes of a 10^{-3} subsystem requirement [Ref. 1]. The 24 events are to be entered sequentially and separately, and there is to be no electrical reset capability for either stronglink switch.[9] The relevance of mathematical treatment of these parameters will be addressed in Sections 2 and 3.

1.2. Imprecision of Safety Requirements

Although the requirement for abnormal-environment safety (10^{-6} per credible exposure) is precise, the limits on credibility of an exposure or accident present a mathematical imprecision that affects UQS analysis. The UQS approach can be tailored to any level of credibility through the UQS parameters (*e.g.*, the number of events used).

2. Resistance to Threats through Pattern Design

The UQS methodology provides resistance to various threats that might be present in abnormal environments, in particular significantly reducing vulnerability to non-random threats, while optimizing resistance to random threats. One way of two basic threat-resistant methods utilized in the UQS approach is pattern design[10]. The constituents of pattern design are event balance (discussed in Section 2.1), transition balance (Section 2.2), bi-valued events (Section 2.3), control of extremes (Section 2.4), and randomness (Section 2.5).

2.1 Resistance to Threats through Event-Type Balance

One unique-signal-pattern design principle protects against inadvertent threats whether or not they are statistically biased toward one or more event values (where each event value has a "type" chosen from a population *s*) by balancing the appearance of each event type.

First consider a UQS pattern having *r* events, each chosen from two event types, *A* and *B*, where the number of *A*s in the pattern is greater than the number of *B*s. If a (highly hypothetical) fault environment were to occur such that events were generated independently and $P(A) > ½$, $P(UQS)$ would necessarily be greater than $(1/2)^r$. If a pattern were designed with half *A*s and half *B*s the most threatening independently generated events would have ($P(A) = P(B) = ½$), $[P(UQS)]_{max}$ is $(½)^r$, which offers a firm probability bound for the conditions given. This can be proved in a variety of ways. For illustration, let $r = 2$ and $s = 2$ (*e.g.*, a pattern with one *A* and one *B*):

---

[8] George Novotny was instrumental in formulating and incorporating this concept.

[9] The MC 2969 was the first SL, and its pattern and reset capabilities were necessitated by contemporary operational constraints.

[10] The other is communication technique (see Section 3).

$$P(A,B) = P(A)P(B) = P(A)[1 - P(A)] = P(A) - P(A)^2$$

Taking the derivative, $1 - 2P(A) = 0$, $P(A) = ½$, and $P(A,B)_{max} = ½x½ = ¼$ (1)

A proof for any r is given in Ref. 2. The conclusion is: **The optimum resistance to an inadvertent independent-event threat, whether or not the threat is statistically biased toward one event type or the other (*e.g.*, *A*s or *B*s), is achieved by using a UQS pattern of balanced (*r*/*s*, as close as possible) event types** (*e.g.*, 12 *A*s and 12 *B*s).

2.2. Resistance to Threats through Transition Balance

Inadvertently received events could not be assured to be independent. Since dependence between successive inputs (first-order transitions) is the most fundamental form of dependence, it is the basis for a second principle for UQS pattern design. **The optimum resistance to an inadvertent threat whether or not it is statistically biased toward one or more of the s^2 first-order transition (*e.g.*, *AA*s, *AB*s, *BA*s, *BB*s for $s = 2$) is achieved by using a UQS pattern of $\frac{r-1}{s^2}$ (as close as possible) of each possible transition pair** (*e.g.*, 6 *AA*s, 6 *AB*s, 6 *BA*s, and 5 *BB*s).[11] Here, r is the number of events and s is the number of event types.

A possible extension for 24 bi-valued events ($r = 24$; $s = 2$) could be to balance members of the set of second-order transition trios (*e.g.*, *A*|*AA*, *B*|*AA*, *A*|*AB*, *B*|*AB*, *A*|*BA*, *B*|*BA*, *A*|*BB*, *B*|*BB*, where | denotes "conditional on") by using a UQS pattern containing as nearly as possible 1/8 *AAA*s, 1/8 *AAB*s, 1/8 *ABA*s, 1/8 *ABB*s, 1/8 *BAA*s, 1/8 *BAB*s, 1/8 *BBA*s, 1/8 *BBB*s. Since there are 22 second-order transitions, the appropriate numbers would be 3, 3, 3, 3, 3, 3, 2, 2, in any order. An example pattern that has all three types of balance (event types, first-order transitions and second-order transitions) is[12]:

E = *A*,*B*,*A*,*A*,*A*,*B*,*B*,*A*,*A*,*A*,*B*,*B*,*B*,*B*,*A*,*B*,*A*,*A*,*B*,*A*,*B*,*B*,*B*,*A* (2)

This could be further extended to any order, denoted by "*d*." However, even balancing second-order transitions has not been a goal of UQS pattern design. One reason is that the family of dependence relations is much more extensive than d^{th}-order transitions [*e.g.*, Ref. 3]. Another reason is that carrying balance beyond first-order dependence tends to make the patterns more susceptible to other types of generation by straightforward computation (discussed in Section 2.7).

2.3. Advantage of Bi-Valued Events

Bi-valued events are necessary for unique signal patterns because of their resistance to dependent threats (inadvertent communication entities that are related to one another),

[11] The first design principle requires an even number of events for optimum resistance; the second requires an odd number of events. The actual practice is to use a multiple of four for the number of bi-valued events (e.g., 24) and to balance first-order transitions as closely as possible (e.g., 6, 6, 6, and 5).

[12] The separation of events by commas is to emphasize that the events are sent sequentially and separately, which is described in Section 3.

and they are readily amenable to stronglink switch implementation as well. The basic mathematical structure for this discussion is based on the probability that one event will be inadvertently generated, conditional on the previous event or events being generated. The formal equations for first- and second-order adjacent dependence are:

$$P(E_i \cap E_{i+1}) = P(E_i) \times P(E_{i+1} \mid E_i) \tag{3}$$

$$P(E_i \cap E_{i+1} \cap E_{i+2}) = P(E_i) \times P(E_{i+1} \mid E_i) \times P(E_{i+2} \mid E_{i+1}) = P(E_i \cap E_{i+1}) \times P(E_{i+2} \mid [E_i \cap E_{i+1}]) \tag{4}$$

where $\cap$ indicates logical "and," and | indicates "conditional on." Eqs. 3 and 4 account for two types of dependence, where an event can depend on one or two preceding events. The reason for resistance to dependence can be demonstrated by an example calculation. Three patterns (one having two event types, one having three event types, and one having four event types), all of which meet the conditions of balanced events, balanced first-order transitions, and balanced second-order transitions (where the balance is optimized), are given in Table 1:

Table 1. Illustrative Patterns of Various Numbers of Event Types

Number of Event Types	*Pattern*
2	*A,B,A,A,A,B,B,A,A,A,B,B,B,B,A,B,A,A,B,A,B,B,B,A*
3	*A,B,A,A,C,B,B,A,C,C,B,C,A,B,C,C*
4	*B,C,A,D,C,C,B,A,B,D,D,A*

These three patterns are all very nearly equivalent in terms of response to independent event threats and each has optimum design features for resistance to first-order and second-order adjacent dependence threats. However, the amount of resistance to these dependence threats differs greatly, as shown in Fig. 1. The results shown for first- and second-order dependence were derived by using a form of "P_{calc}" [Ref. 1] that has been generalized here:

$$P_{calcg} = \left[\left(\frac{c_{11}}{\sum_{i=1}^{s} c_{1i}}\right)^{c_{11}}\left(\frac{c_{12}}{\sum_{i=1}^{s} c_{1i}}\right)^{c_{12}} \cdots \left(\frac{c_{1s}}{\sum_{i=1}^{s} c_{1i}}\right)^{c_{1s}}\right]\left[\left(\frac{c_{21}}{\sum_{i=1}^{s} c_{2i}}\right)^{c_{21}}\left(\frac{c_{22}}{\sum_{i=1}^{s} c_{2i}}\right)^{c_{22}} \cdots \left(\frac{c_{2s}}{\sum_{i=1}^{s} c_{2i}}\right)^{c_{2s}}\right] \cdots \left[\left(\frac{c_{s^d 1}}{\sum_{i=1}^{s} c_{s^d i}}\right)^{c_{s^d 1}}\left(\frac{c_{s^d 2}}{\sum_{i=1}^{s} c_{s^d i}}\right)^{c_{s^d 2}} \cdots \left(\frac{c_{s^d s}}{\sum_{i=1}^{s} c_{s^d i}}\right)^{c_{s^d s}}\right] \tag{5}$$

where "c" is the number of appearances for each event or event combination on which the transition is conditional, and "d" is the (non-zero) order of adjacent dependence. The general form shown in Eq. 5 matches the frequency of occurrence of combinations in the

pattern of any particular order of adjacent dependence for the maximum threat of that order, following the first *d* events.

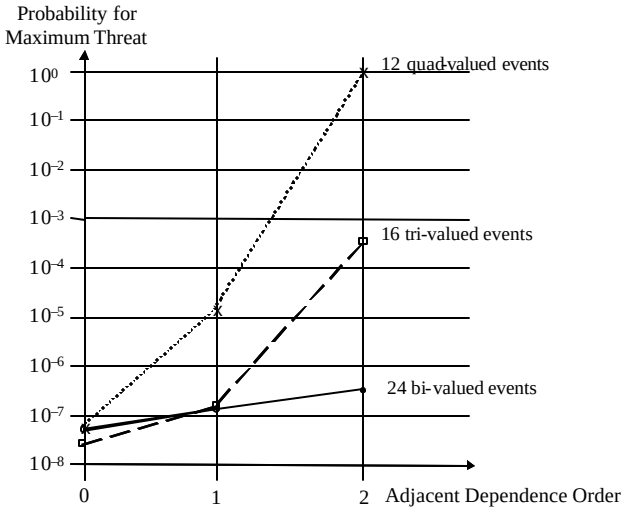

Figure 1. Dependence Effects as a Function of Number of Event Types

In Fig. 1, zero dependence order is equivalent to independence, adjacent first-order dependence means each event is influenced by the immediate previous event, and adjacent second order dependence means each event is influenced by the previous two events[13]. The number of events used in each pattern makes all three patterns approximately equivalent for independent threats. Higher orders of dependence show rapid degradation as the numbers of event types increase. The results in Fig. 1 illustrate the basic advantage of bi-valued events; they provide greater protection against dependence effects.

In the late 1980s, Curt Mueller demonstrated a similar effect, producing the relationship shown in Fig. 2. The abscissa represents the number of event types, *s*, and the ordinate represents the number of events that would be required to match (as closely as possible) the first-order-adjacent-dependence P_{calc} result for the MC 2969 stronglink switch. The P_{calc} result was obtained by the equation:

$$P_{calc} = \left(\frac{w}{w+x}\right)^{w}\left(\frac{x}{w+x}\right)^{x}\left(\frac{y}{y+z}\right)^{y}\left(\frac{z}{y+z}\right)^{z} = \left(\frac{32}{36}\right)^{32}\left(\frac{4}{36}\right)^{4}\left(\frac{7}{10}\right)^{7}\left(\frac{3}{10}\right)^{3}$$
$$= 7.8\times 10^{-9} \qquad (6)$$

where *w* is the number of *A*s followed by *A*s, *x* is the number of *A*s followed by *B*s, *y* is the number of *B*s followed by *A*s, and *z* is the number of *B*s followed by *B*s. Since a balanced pattern would require fewer events for the same level of protection, the

[13] For nonzero adjacent dependence order, P_{calc} begins with the assumption that the first "*d*" events have already occurred.

calculation for $s = 2$ uses Eq. 6 with $w = x = y = 7$; $z = 6$ (28 events). Higher values of s require similar application of the more general P_{calcg} solution given in Eq. 5.

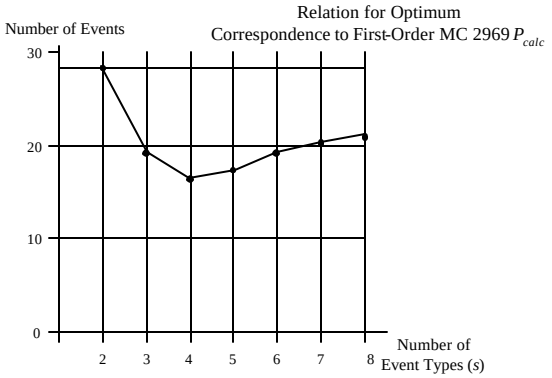

Figure 2. A Restrictive Model for the Effect of Number of Event Types

The results showed improvement in event efficiency up to $s = 4$, and then degradation for higher values. The curve does not monotonically decrease because first-order-adjacent dependence introduces an effect counter to the efficiency improvement that independent events would offer. These results, along with the additional effects shown in Fig. 1 help demonstrate why **the UQS pattern-design portion (as well as the communication portion) of dependence resistance is optimized if the number of event types is two.**

2.4 Reduction of Likelihood of Extremes

As a contributor to the methodology (along with communication technique), the pattern of event types is an important factor in the independent behavior sought. The pattern also contributes to the narrow variance sought in pattern uncertainty (resistance to threats having extreme values). A key to understanding how the pattern relates to uncertainty is provided mathematically. Consider a pattern of r events, all of one type:

Example pattern: $A,A, \ldots,A$

A threat that was statistically biased toward all As would result in loss of safety. This leads to the requirement that the pattern used have r/s of each event type (as close as possible, if s does not divide r). The optimum choice is for r to be a multiple of four (*e.g.*, 24), and for s to be 2, but the results given here are not limited to those values.

Further enhancements in approaching independent events and minimal variance in uncertainty require that first-order transitions (of which there are s^2) be balanced, as closely as possible. This means that the pattern should contain (as close as possible) $\frac{r}{s^2}$ of each transition pair. For example, if $s = 2$, and representations A and B are used, there are four transition pairs ([A,A], [A,B], [B,A], and [B,B]). If r were 24, there would be 6, 6, 6, and 5 (in any order) of each transition pair. In general, there are $r - d$ transitions for d^{th}-order transition groups. The total number of transitions of order d is s^{d+1}, so there is a

limit to the practicality of balancing ($\frac{r-d}{s^{d+1}}$ of each transition group) within the r events. An example balanced (for d = 1 and 2) pattern for which s = 2 and r = 24, which was first shown in Eq. 2, is the best balance that can be achieved for numbers of events of each type, for first-order adjacent transitions, and for second-order adjacent transitions. Balance for higher order of d than 2 is meaningless for $s > 2$ and r = 24, because for these values, $r - d$ cannot be meaningfully divided by s^{d+1}. Even balance for order two is not sought in modern UQS pattern design.

Theoretically, there could be s^r different potential patterns, and one might naively expect that the probability of randomly compromising a "correct" combination would be s^{-r}. However, there are two major considerations that introduce further constraints. One is that there are threats that can bypass inadvertent information duplication. Not only must the subsystem threat be shared, but also the UQS portion provides the best opportunity for protection. Fig. 3 depicts a fault tree that includes environmental threats along with the information threat that affects the UQS. Since it is difficult to assure that environmental bypass risk is much less than the intent or trajectory system requirement, the UQS portion must be *far* safer than the requirement.

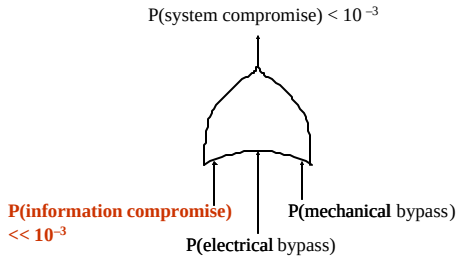

Figure 3. Safety Bypass Fault Tree

A probabilistic equation for this is:

$$P(B) = P(I) + P(M)[1 - P(I)] + P(E)[1 - P(I)][1 - P(M)] \quad (7)$$

where $P(B)$ is the probability of inadvertent bypass, $P(I)$ is the probability of inadvertent information generation, $P(E)$ is the probability of inadvertent electrical bypass, and $P(M)$ is the probability of mechanical bypass. This assumes $P(I)$, $P(E)$, and $P(M)$ are independent.

If $P(I)$ were s^{-r}, then $P(B) > s^{-r}$. $P(M)$ and $P(E)$ should be minimized, for example by using a robust stronglink switch and co-located weaklinks. This point helps illuminate how the stronglink switch isolation and inoperability must be part of the indicated subsystem probability calculation. The stronglink switch can be tailored to the expected environments, protected by the barrierization of the stronglink switch package and the implementation of the weaklinks. But there are practical limits on how far this can be carried, and since the UQS protection is not so limited, it can provide the most effective subsystem protection, with $P(I) << s^{-r}$.

A second major consideration is that probabilities represent an average expectation that may rarely be met. For example, if an integrated circuit component yield were 0.35, the statistical expectation that there would be exactly 35 successes in a run of independent tests of 100 components is less than 10%. Another important parameter is how great the *spread* is from the average expectation. One of the major contributions to the UQS methodology is that the spread can be reduced by judicious implementation choices, which is another example of statistical-bias-resistance. From a safety viewpoint, this means that extreme deviations from the mean toward higher probabilities can be reduced. The main methods of control are:

1. To approach a particular goal, the parameters s and r can be controlled. In order to achieve a spread of values of $P(I) << 10^{-3}$, s should be equal to two and r should be equal to about 24 [Ref. 2]. The reason for $s = 2$ is that higher values of s create more vulnerability to dependence, and therefore more spread. The amount of spread cannot be determined exactly as a formal result, but the derivations above indicate the advantage obtained, and it can be clearly illustrated by a constrained example. Compare two patterns, one with $s = 4$ and $r = 4$, and the other with $s = 2$ and $r = 8$. For the first, an optimized pattern is *ABCD* and for the second, an optimized pattern is *ABBBABAA*. Both of these patterns have optimum event frequencies and transition frequencies for the given values of s and r. If the pattern events were equally likely and independent, the probability of inadvertent duplication would be 1/256 in both cases. But there can be statistically biases, and one event can have residual dependence on another, in spite of all efforts to remove dependence. Under these assumptions, the maximum threat to the first pattern is when (with *A*, *B*, *C*, and *D* equally likely) $P(B|A) = P(C|B) = P(D|C) = 1$. The probability is $¼ \times 1 = ¼$. The maximum threat to the second pattern under the same assumptions is (with *A* and *B* equally likely) $P(A|A) = 1/3$, $P(B|A) = 2/3$, $P(A|B) = 1/2$, and $P(B|B) = 1/2$ (matching transition frequencies in the pattern), for which the calculated pattern probability is $½ \times 1/3 \times 2/3 \times (½)^4 = 1/216$. Therefore, the maximum probability is reduced by a factor of 54 by using the second pattern instead of the first, although the "random" result is the same. **This is indicative of why spread from the mean is minimized by using fewer event types**, exemplified by the second pattern. This strategy mitigates extreme threat levels.
2. The r events must be sent (insofar as practical) separately and unrelated to each other. This is to resist various forms of non-random threats. An illustration of potential resistance is to consider a "roulette-like" wheel with four sectors[14]. For a single spin, the probability of the pointer signifying a particular sector out of the four possible cannot be bounded without knowing the wheel balance and sector sizes. However, for a wheel with two sectors that is spun twice, the probability of signifying a particular sector first and

[14] More details on this analogy are given in Section 3.3.1.

then the other (one of four possibilities) cannot be greater than ¼ (see Eq. 1 derivation). **By constructing and sending each event separately, and by designing an appropriate pattern, any statistical bias toward the correct pattern must occur *r* times instead of once.** The separate reception strategy is also analogous to a person scanning the text of a report. The person would see considerable dependence from letter to letter, but if their read process were constrained to not follow every letter (causing skipped text before reading the next letter), the selections would be more independent.

2.5 Randomness Metrics

In order to resist non-random threats as well as random threats, each UQS pattern (and the set of UQS patterns) must meet randomness requirements. First the randomness of individual patterns will be addressed. The requirements for joint randomness will be addressed in Section 2.6. The randomness property of a sequence means that there should be no predictability of a sequence member, given any information about previous sequence members, with the exception that the next result can be restricted to a set of possible members (*e.g.*, the set of event types). This is a desirable characteristic for UQS patterns, because non-predictability implies difficulty of inadvertent generation.

2.5.1. Theory of Runs

A Bernoulli process has randomness characteristics because it requires *n* independent equally likely binary choices [*e.g.*, Ref. 4]. We define *runs* as strings of arbitrary length of identical values. One pertinent metric is the number of such runs in a unique signal pattern. The theory of runs[15], derived from the Bernoulli process [Ref. 5], demonstrates that in a sequence of *n* events, the mean number of runs for a Bernoulli process is:

$$m = \frac{2n_A n_B}{n_A + n_B} + 1 \tag{8}$$

where n_A is the number of *A*s in the sequence, and n_B is the number of *B*s. The variance is:

$$s^2 = \frac{2(n_A n_B)^2}{(n_A + n_B)^2 (n_A + n_B + 1)} \tag{9}$$

Table 2. Measures of Runs

Name	Pattern	Number of runs
E-example (E)	*A,B,A,A,A,B,B,A,A,A,B,B,B,B,A,B,A,A,B,A,B,B,B,A*	13
C-module (C)	*A,B,B,B,B,A,A,A,B,A,A,A,B,B,A,A,B,B,B,A,B,A,A,B*	12
D-module (D)	*A,B,A,A,A,A,B,A,A,B,A,A,B,B,B,B,A,B,B,B,B,A,A,B*	12
F-example (F)	*A,B,A,A,A,A,B,A,A,B,A,B,B,A,A,B,B,B,B,B,A,A,B,B*	12
Oscillatory (O)	*A,B,A,B,A,B,A,B,A,B,A,B,A,B,A,B,A,B,A,B,A,B,A,B*	24

[15] Bob Thompson suggested examination of this metric in the 1980s time frame.

The number of runs in UQS patterns shouldn't differ greatly from the Bernoulli statistics, although a large variance is indicative of some leeway.

For 12 *A*s and 12 *B*s, this metric suggests there should be approximately 13 runs, but the variance is a little greater than three. The example in Eq. 3 had 13 runs. There are 12 runs in modern 24-event unique signal patterns. A pattern of alternating *A*s and *B*s would have 24 runs (the maximum possible). This is part of the information tabulated in Table 2. The minimum number possible is two, if there are at least two event types.

The first table row is the example pattern (E) given in Eq. 2. The second row is the modern "Intent stronglink switch" pattern, developed for the "C-module" stronglink switch. The third row is the "Trajectory stronglink switch" pattern, developed for the "D-module" stronglink switch. The fourth row is another example pattern (F) that will be discussed in Section 2.7. The fifth row shows an oscillating pattern (O) for contrast.

2.5.2. Run Frequencies

If the number of runs n_A and n_B of length k are tabulated as n_{Ak} and n_{Bk}, the randomness expectation for Bernoulli processes is that $n_{Ak} \approx n_{Bk}$ for each k, and that $n_{Ak} \approx 2n_{A2k}$. Table 3 compiles these results for the same patterns shown in Table 2.

Table 3. Run Frequencies

Pattern	n_{A1}	n_{B1}	n_{A2}	n_{B2}	n_{A3}	n_{B3}	n_{A4}	n_{B4}
E	4	3	1	1	2	1	0	1
C	2	3	2	1	2	1	0	1
D	2	4	3	0	0	0	1	2
F	2	3	3	2	0	0	1	0
O	12	12	0	0	0	0	0	0
M	3	3	1.5	1.5	1	1	0.75	0.75

The last row (M) shows the mean Bernoulli expectation for reference. All entries in the first two rows (the example pattern and the C-module pattern) are within one of the Bernoulli numbers. The numbers in the third and fourth rows are within two. The fifth row (alternating *A*s and *B*s) would be expected to deviate significantly. Even if the Bernoulli statistics were matched exactly by the runs count (which would be impossible since runs must be integers), there would still be a concern for UQS patterns. This is because the regularity of having the statistics for *A*s match those for *B*s exactly are vulnerable to abnormal-environment-caused algorithmic generation. This will be elaborated on in Section 2.7.

2.5.3. R-Statistic

The *R*-Statistic [Ref. 6][16] counts sequence appearance positions for events of each type and compares by summing the absolute values of the differences. This number for UQS patterns should also approximately match Bernoulli statistics. The metric is:

$$R = \sum_{i=1}^{12} | a_i - b_i | \tag{10}$$

where a_i is the sequence position of the i^{th} *a*, and b_i is the sequence position of the i^{th} *b*. Some example results of this metric are shown in Table 4.

Table 4. Examples of *R*-Statistic Metric

Pattern	R-Statistic Metric
E	34
C	22
D	60
F	62
O	12

Although tighter bounds could have been sought, the general UQS pattern goal is between about 20 and 60. The result shown for the oscillating pattern "O" is the minimum that can be achieved; 144 is the maximum (12 *A*s followed by 12 *B*s).

2.5.4. Autocorrelation

A pattern having maximum uncertainty would be expected to rapidly decrease in autocorrelation. An example autocorrelation function, *c*, that is applicable to the discrete, bi-valued, truncated UQS patterns is:

$$c_j = \sum_{j=0}^{24} \frac{1}{24 - j} \sum_{i=1}^{24-j} a_i a_{i+j} \tag{11}$$

where $a_i = -1$ for an *A* event and $a_i = 1$ for a *B* event. The autocorrelation functions for the examples used previously are shown in Fig. 4. These results indicate that the first four patterns being illustrated (E, C, D, F) have reasonable autocorrelation functions, but the fifth (Oscillating) does not. Judging "reasonableness" for autocorrelation functions is highly subjective, although improvement can be obtained by transform techniques, such as Fourier transforms[17]. One of the reasons for the difficulty is that higher abscissa values have fewer components under the right–most summation, and therefore more chance for the ordinate to deviate. However, the metric is a useful general indicator.

[16] This metric was suggested to Jackie Leyland by Peter Mueller and Brani Vidakovic, Institute of Statistics and Decision Sciences, in the mid 1990s, during her M.S. work at Duke University.

[17] This was technique was suggested by Ken Chen.

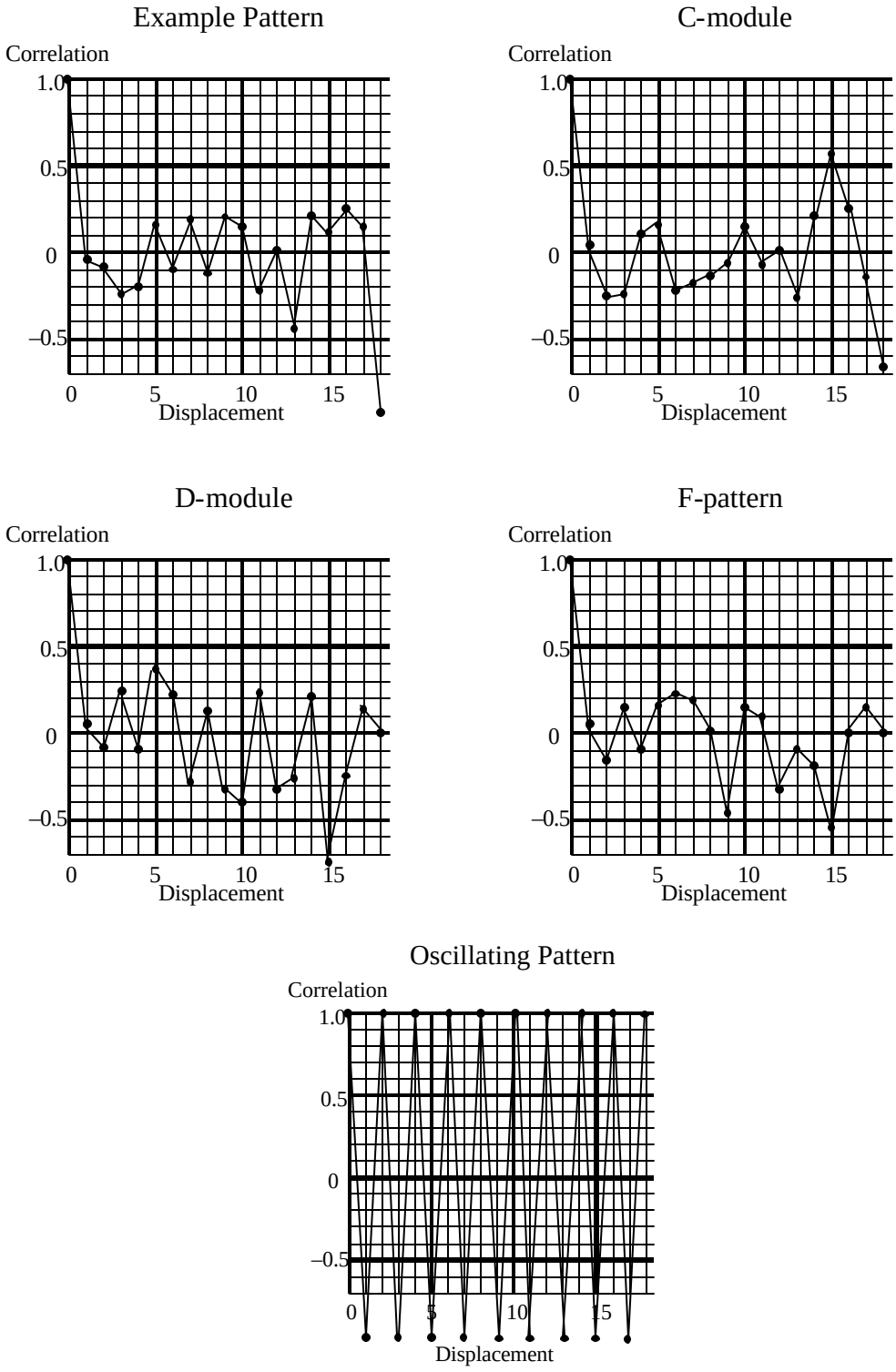

Figure 4. Autocorrelation Functions for Selected Patterns

2.5.5. Entropy

The information term "entropy" (high entropy indicates disorganization that reduces ease of identification of the most likely messages) can be adapted as a measure of uncertainty, where an entropy value of one corresponds to maximum entropy (and therefore maximum uncertainty). The formal definition of bi-valued information entropy is a modified form of traditional entropy [Ref. 7]:

$$H = -\left(\sum_i p_i \log_2 p_i \right) / n \tag{12}$$

where the summation is over the potential messages, each having a probability p_i, and n is the length of the messages (in bits).

Applying the entropy concept to bi-valued UQS patterns requires defining a number of n-character "entities," indexed by i, that can be single events or groups of events. The probability of each of these is derived from the maximum threat probability, which corresponds to matching the frequency of appearance of the entities. For example, a pattern with 12 *A*s and 12 *B*s would have $p_1 = p_2 = ½$, resulting in unit entropy. The four transition pairs in modern UQS patterns have an entropy metric of 0.998. The eight transition trios of the example pattern of Eq. 2 with balanced trios have an entropy metric of 0.994.

Table 5 shows the entropy metrics for transition pairs and trios for the examples that are being demonstrated.

Table 5. Example Entropy Metrics

Pattern	Transition pairs	Transition trios
E	0.998	0.994
C	0.998	0.967
D	0.998	0.960
F	0.998	0.967
O	0.499	0.333

The first four patterns (E, C, D, F) have optimal transition-pair entropy because all of these have balanced transition pairs. The first pattern (E) has optimal transition-trio entropy because it has balanced transition trios. The entropy metrics for the oscillating pattern are very low, as would be expected for a pattern of such regularity.

2.5.6. Theory of Linear Complexity

Another form of randomness metric is determination of the length of the shortest linear feedback shift register (LFSR) that can generate the pattern sequence.[18] This is referred to in cryptology as "linear complexity." A generalization to non-linear shift registers is also of interest in UQS safety analysis. On the average, replication of binary sequences requires $n/2$ LFSR stages. Some sequences that appear random and meet many other randomness tests can have a linear complexity metric of $log_2 n$. For maximum uncertainty, the goal for UQS patterns is for both the LFSR and the nonlinear feedback shift register metrics to approach n.

2.6. The Utility of Randomness Metrics

Randomness metrics are useful as some specific forms of guidance, but are not used exclusively in UQS pattern selection. Curt Mueller once offered the analogy: "If you see and dispose of three cockroaches on your kitchen floor, would you assume there were no others in your house that you hadn't found?" This is a good message for anyone tempted to base all abnormal-environment assessment on a few mathematics models. Other mathematically based, but subjectively applied criteria are that there should be (in addition to balanced *A*s and *B*s and balanced transition pairs):

1. At least one run of length one for each event type, with no runs longer than four.
2. No long (*e.g.*, 6) duplicated sub-patterns or inverses (complementing event types) anywhere within the pattern.
3. No long (*e.g.*, 8) mirrored (reverse order) or inverse mirrored sub-patterns anywhere within the pattern.
4. Dissimilar run characteristics for the two event types.
5. No significant oscillatory portions to patterns.

UQS patterns have a potential population of 2^{24} patterns (24 bi-valued events), but only several dozen pass the initial screening tests. Since all pairs of patterns must meet similar additional constraints in order that the presence of one pattern is unrelated to any other, the number of patterns that can actually be used in weapons safety applications is less than two dozen. For pairs of patterns, some other constraints are that there should be:

1. No long (*e.g.*, 9) common or inverse duplicated sub-patterns.
2. No long (*e.g.*, 5) aligned (same position in sequence) common or inverse sub-patterns.
3. No more than 15 nor fewer than 9 aligned event type matches.

Part of the reason for these metrics is that complete dependence on mathematical analysis misses the contribution of human insight. During the selection of UQS patterns, experienced personnel look for bothersome features and can reject a pattern on the basis of discomfort alone. While an argument could be made that any human-developed

[18] Jim Davis pointed out this approach in about the 1985 time frame.

feelings could be programmed, it is instructive to remember that it is extremely difficult to mathematically forecast all abnormal environments.

2.7. Insufficiency of Randomness Metrics

Mathematical randomness metrics are useful, but they are transcended by human judgment. A pattern that meets all known mathematical tests would not be appropriate if it were easily generated by some common process, where such a process had been designed or could be easily implemented by inadvertent re-configuration due to abnormal-environment changes. It was pointed out earlier that identical uncertainty characteristics for both (or all) event types should be avoided. One of the reasons is that this "sameness" is a characteristic of pseudorandom generators and other similar straightforward implementations, which are commonly used and easily constructed (intentionally or inadvertently).

The basic mathematical structure utilized for pseudorandom generators and for the one-way transforms to be examined in Section 4.2 is detailed in Ref. 8 and summarized in Appendix E. Pseudorandom generators are based on Galois field structures and operations, which are implemented, for example, using "exclusive-or" logic gates and "and" gates, as LFSRs (linear feedback shift registers).

Consider the generator polynomial for the recursive relationship $x_{i+5} = x_{i+2} + x_i$, which is:

$$x^5 + x^2 + 1 = 0 \tag{13}$$

This is one of the most commonly used fifth-degree polynomials, and it has the smallest degree that can generate reasonable 24-bit patterns. The polynomial in Eq. 13 (and the recursive relationship) corresponds to the logic circuit shown in Fig. 5. The operations shown are shift (divide by x) and exclusive-or.

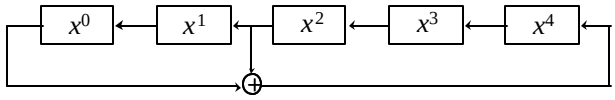

Figure 5. Logic Circuit Corresponding to Generator Polynomial $x^5 + x^2 + 1 = 0$

Starting with $x^1 = 1$, and $x^0 = x^2 = x^3 = x^4 = 0$, the sequence generated out of x^0 on division by x is indicated by the sequence of ones and zeros below:

0,1,0,0,0,0,1,0,0,1,0,1,1,0,0,1,1,1,1,1,0,0,0,1,1,0,1,1,1,0,1
A,B,A,A,A,A,B,A,A,B,A,B,B,A,A,B,B,B,B,B,A,A,B,B

Underneath this sequence is repeated the "F-example" used earlier. Here, "F" stands for "failing," because the first 24 bits of the pseudorandom generator pattern differ in only

the next-to-last position from the F-example pattern, which makes it extremely straightforward to generate a near-match. And yet, the F-example is balanced in terms of *A*s and *B*s and in terms of transition pairs. It has 12 runs, as do modern UQS patterns. It is within two of the Bernoulli run statistics, as is the D-module pattern. It has an *R*-statistic of 62, only slightly different than the 60 for the D-module pattern. It has a comparable autocorrelation pattern to modern UQS patterns. Its entropy is the same as the C-module pattern.

Pseudorandom signals satisfy basic randomness tests, but do not have satisfactory uncertainty characteristics. Examining the 31-bit LFSR sequence, there are four runs of a single one and four runs of a single zero. There are two runs of a pair of ones and two runs of a pair of zeros. There is one run of three ones and one run of three zeros. This symmetry is lost only slightly at the extremes, where there is one run of four zeros and one run of five ones. From the standpoint of designers' use of pseudorandom generators, this event-type sameness and lack of uncertainty yields the benefit of straightforward circuitry. From the viewpoint of safety assessment, sameness is commensurate with ease of threat generation. This helps demonstrate why engineering judgment is important in conjunction with mathematical tests of randomness.

3. Resistance to Threats through Separate-Event Communication

Separate-event communication is a key feature of UQS methodology. **The basic reasons for separate-event communication are to increase resistance to potential abnormal-environment biases and to reduce dependence effects**.

3.1. Existence of Dependent Information Constituents

The UQS pattern constraints (*e.g.*, to balance event types and numbers of transition pairs) contribute to resistance to dependence-related bias, so the pattern design coupled with its sequential communication provides a clear advantage over other techniques, such as the recognition of combinations. In fact, it is the manner in which the UQS pattern develops sequentially that offers resistance to the types of sources (*e.g.*, oscillators and LFSRs) that could be easily coupled to or provided as inputs for stronglink switches. These relations between inadvertently generated events cannot be precluded, but **resistance can be provided by the stronglink switch response (one event at a time, sequentially and separately). This stronglink switch function must be supported throughout the communication system for maximum effectiveness**.

Dependent processes can have similarities to or differences from any UQS pattern. Although this sort of threat statistically bias is unavoidable, **dependent bias can be prevented from degrading safety if events are communicated separately**. Here, dependent processes will be demonstrated as a means of identifying the problem, in order to help emphasize that separate-event communication is a necessary feature of UQS methodology.

A dependent process (*e.g.*, Ref. 9) is a relation between the probability of an event type and a previous event type or types.

$$P(E_i) = P(E_i \mid E_j, E_k, ...) \tag{14}$$

3.1.1. Oscillatory Dependence

Modern UQS patterns are carefully designed to minimize vulnerability to oscillations, but the vulnerability cannot be eliminated without using separate-event communication. As an illustration, consider the mathematical combination of three oscillations:

$$y = \sin(2.28x - \frac{p}{11}) + 0.855\sin(0.254x - \frac{7p}{9}) + 0.26\sin(1.1x - \frac{p}{8}) \tag{15}$$

where samples are taken at x = 0, 1, 2, … , 23. Assume that an *A* is generated if $y < 0$, and a *B* is generated otherwise. The result is the D-module UQS pattern[19].

Eq. 15 demonstrates an additive combination of oscillatory sources. Some other of many possible examples are multiplicative cycles (a cycle of length three and one of length five combine to give a cycle length of their least-common-multiple, fifteen), switched sources (changes during the sampling period), and modulated signals.

3.1.2. Physical Measurement

In a weapon environment, many related measurements are possible, and therefore information that might be inadvertently coupled to UQS communication is a potential source of dependence. Measurements come from various sensors, e.g., temperature, pressure, velocity, stress, time, and position. Any such dependent collection of information could simulate and be statistically biased toward a UQS pattern, which necessitates separate-event communication.

3.1.3. Logic Relations

Logical processing for reliability functions, such as parity checks, CRCs, and error-protection coding cause dependence, as do many ordinary processing functions, such as counters, headers, shift registers, and transformation generators.

Communication coding, such as ASCII, BCD, and Baudot representations preclude independent information. These not only constrain the potential population, but also do it in a structured manner (*e.g.*, including parity checks to assure an odd or even number of bits in a word) that adds dependence.

As an example of a constrained effect, consider a threat population consisting of all possible combinations of 12 *A*s and 12 *B*s[20]. Under random selection from this population, the probability of producing any particular modern UQS pattern is[21]:

[19] This was developed in Ref. 3 with the aid of Rademacher-Walsh expansion [Ref. 10].

$$P(UQS) = \binom{24}{12}^{-1} = 3.7 \times 10^{-7} \tag{16}$$

Here, the threat population is reduced by more than a factor of six. Again, the dependence provides a reason for separate-event communication.

3.2. Prevention of Encryption, Error-Protection Coding, and Transformations

Since the UQS pattern is an essential part of its abnormal-environment protection, **any potential perturbations to the pattern must be avoided. This means that the pattern cannot be encrypted, protected by error-correcting codes, re-ordered, or transformed in any way**. The separate-event communication technique helps discourage applying these types of communication strategies to UQS patterns.

3.3. Prevention of Dependence Effects

Since the abnormal-environment threat cannot be controlled, it might be tempting to think that nothing can be done about the dependence threat. However, separate-event communication decreases both the effects of dependence and the amount of variance expected (extent of extreme effects). Although both the UQS pattern and its communication are necessary to the UQS methodology, only the communication technique will be addressed in this paragraph. An informative communication analogy was proposed by Stan Spray in the late 1980s. Assume a critical abnormal-environment-resistant message is to be transmitted from a control point to a receiver by telephone. The message chosen is "Albuquerque."[22] The threat is that cross-communication will couple unintended telephone conversations to the receiver, and that the word "Albuquerque" might be received inadvertently as an unintended control message. The fact that "Albuquerque" might be used in conversations cannot be avoided. But by placing eleven calls rather than one, each containing one letter (first "A," then "l," then "b," etc.), the inadvertent coupling must occur eleven times rather than once. The example message would not be a suitable UQS pattern, for example because it has seven event types, but the example helps illustrate how separate-letter communication is less resistant to the dependence inherent in words such as "Albuquerque" than is single-word communication. A mathematical demonstration follows.

3.3.1. Advantages of Communication in Conjunction with Pattern

The following results, expanding on the wheel example in Section 2.4, help show the advantage of separate-event communication in reducing variance and therefore statistical

---

[20] Systems that increase sensitivity by biasing to a mean value of fluctuations and by "paged" memories have a tendency toward equal kinds of values.

[21] This scenario was developed by Curt Mueller in the late 1980s. Note that $\binom{24}{12} = \frac{24!}{12!(24-12)!}$

[22] Although the choice of "Albuquerque" was parochial, the basic idea is unaffected by the word choice. Whatever the word chosen is, there is no way to preclude it or to reliably estimate its frequency of use.

bias toward extremes. The example that will be mathematically demonstrated is counterintuitive to almost everyone who sees it for the first time. Consider four bins, in each of which is placed a numeric value randomly selected from the set {0, 1, 2, 3}, where replacement for the set is assumed. This is analogous to selection of values from a stationary (non-varying properties) communication channel, where the selection may be once (single message communication) or twice (two-message communication). First, select a bin at random, and read (with replacement) the randomly selected value contained in the bin. The probability of getting any particular value (0, 1, 2, or 3) is ¼. As a physical model, consider a roulette-like wheel with four evenly sized sectors, each sector containing a randomly selected value from the set, which is perfectly balanced and fairly spun. This part of the problem is *not* counterintuitive.

Now consider the same randomly loaded bins, but select randomly on two sequential draws (with replacement) from the bins to obtain two values. (For the spinning wheel, this would require two separate spins of the wheel.) The values will be even (0 or 2) with probability ½ or odd (1 or 3) with probability ½. But the probability of getting an odd value on the first selection and an even value on the second selection (or an even value on the first selection followed by an odd value on the second selection) is not ¼, it is $\frac{3}{16}$ (see derivation below). The probability of getting an odd value followed by an odd value (or an even value followed by an even value) is $\frac{5}{16}$. On reflection, this counterintuitive result is due to a subtle dependence between the two values selected. The two-message communication in conjunction with a properly designed pattern (satisfied by one value on the first selection and another value on the second selection) gives a safer expected value than single-message communication. This generalizes similarly to 24 messages for the 24 events in a modern UQS pattern, and demonstrates one advantage of separate-event communication.

The derivation of the results for two-message communication follows:

There are 16 equally likely patterns of odds and evens that can be loaded into the bins (or sectors). One of the combinations (all odds) results in probability one ($P(a)$) of selecting an odd followed by an odd. The four combinations $\binom{4}{1}$ with a single odd and three evens result in probability ($P(b)$) $\frac{1}{16}$ of selecting an odd followed by an odd. The six $\binom{4}{2}$ combinations with two odds and two evens result in probability ($P(c)$) ¼ of selecting an odd followed by an odd. The four $\binom{4}{3}$ combinations with three odds and one even result in probability ($P(d)$) $\frac{9}{16}$ of selecting an odd followed by an odd. The result for the probability ($P(o, o)$) of selecting an odd followed by an odd (and the probability of selecting an even followed by an even) is:

$$P(o,o) = P(e,e) = \frac{1}{16}P(a) + \frac{1}{4}P(b) + \frac{3}{8}P(c) + \frac{1}{4}P(d) = \frac{5}{16} \tag{17}$$

A similar calculation for an odd followed by and even (or an even followed by an odd) is:

$$P(o,e) = P(e,o) = \frac{3}{8}P(e) + \frac{1}{2}P(f) = \frac{3}{8}\times\frac{1}{4} + \frac{1}{2}\times\frac{3}{16} = \frac{3}{16} \tag{18}$$

where $P(e)$ is the probability of selecting one type and then the other where there are two odds and two evens, and $P(f)$ is the probability of one type and then the other when there is one of one type and three of the other.

3.3.2. Reduced Variance in Separate-Event Communication

The reduction in variance was derived in Ref. 2 and shown in Fig. 16 of that reference for the single-message communication along with the two-message communication as portrayed in Eq. 18 in order to show the reduction in variance. This figure is reproduced in Figure 6 with the two-message communication portrayed in Eq. 17 added for completeness.

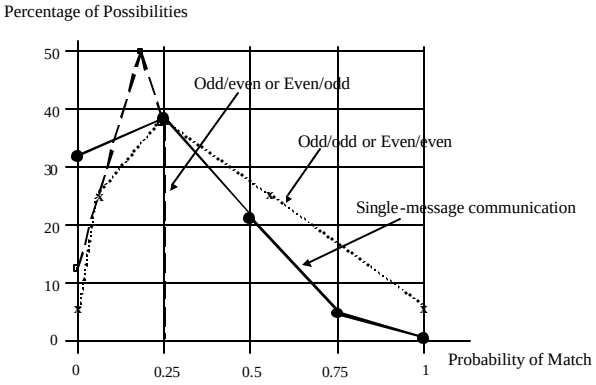

Figure 6. Probability Spreads for Three Communication Strategies

The results are a straightforward demonstration of the advantages of combining separate message communication with careful pattern design. The effective variance of an unrestricted process can be reduced substantially.

3.3.3. Digital Communication Protocol

One of the most important applications of the results described above is to help assure that UQS communication protocol do not inadvertently compromise the effectiveness of the UQS approach. In the mid 1980s, it became clear that digital communication would be used in the future for communication between DoD systems and nuclear weapons. Computer words used in weapons carriers supported up to 32-bit words, and communication protocol supported messages containing large numbers of words. It was tempting for communications engineers to package multiple events within the same

computer word or within the same communication message, which was contrary to UQS methodology.[23]

3.4. Sandia National Laboratories Policy Statement

The effects of digital communications protocol were examined by a large number of Sandia National Laboratories personnel, culminating in a review by five people who were Department Managers at the time (Jim Ney, Glen Otey, Jim Wright, George Merren, and Ray Reynolds). Three people who were Directors at the time (Heinz Schmitt, Herm Mauney, and Gene Ives) prepared a statement in a September 18, 1989 letter to the Chair of the AMC POG specifying how UQS events could be safely communicated in message-oriented communication systems. The statement (pertaining to double-intent[24]) said, in part, **"As a result of a thorough and extensive technical review and evaluation, Sandia National Laboratories has concluded that, in order to meet modern abnormal environment nuclear safety requirements, the 24 events for each intent unique signal need to be sent as one event per message. We believe that this nuclear safety conservative approach to unique signal communication is necessary to minimize the number of safety critical components and to remove the need to analyze the performance of the communication system under abnormal environment situations."**

4. The Care Required in any Processing of Unique Signals

Although UQS events would ideally be communicated separately all the way to a stronglink switch, there are several reasons for signal processing. Since the pattern must not be affected, such processing must be done with care. This section addresses mixing multiple UQS patterns to combine human intent and trajectory information, and one-way transforms and monitoring to test a UQS pattern for correctness.

4.1. Mixing Unique Signal Patterns

Since intent enablement and double intent systems combine human intent UQS patterns with trajectory UQS patterns, event-by-event "mixing" has been used to combinationally produce the trajectory stronglink switch UQS pattern. It was shown 25 years ago[25] that linear (exclusive-or) mixing was superior to non-linear mixing (*e.g.*, "and" function) because equally likely events as inputs produced equally likely events out of the mixing process, as opposed to biasing the output toward the UQS pattern or toward its inverse (which is an inverter away from the correct UQS pattern). This is the reason that all UQS pattern mixing that has been implemented utilizes exclusive-or mixing[26]. It was later shown mathematically [Ref. 11] that nonrandom characteristics were suppressed in

[23] Curt Mueller was the first to point this out. Milt Vernon was the first to bring the need for separate messages to the attention of DoD personnel during System 2 deliberations by the AMAC POG.
[24] The context was explicitly toward two 24-event unique signals, but is implicitly general.
[25] Curt Mueller was the first to show it to the author.
[26] An additional SL could also be utilized inside the exclusion region in a series (logical "and") configuration, but the cost would be substantially greater than that of an exclusive-or gate.

general by mixing. The clarity of the mathematical approach has been improved, as summarized below.

Since exclusive-or is identical to GF(2) (see Appendix E), its properties can be shown for two inputs, which then generalizes to any number of inputs due to the mathematical linearity. Assume that source events "*a*" and source events "*b*" are exclusive-or mixed to yield output events "*c*." The probability of a correct output (matching the intended UQS event output), $P(c)$, as a function of the probability of a correct input (matching one intended UQS event input), $P(a)$, and a correct input (matching another intended UQS event input), $P(b)$, is:

$$P(c) = P(a)P(b) + [1 - P(a)] \times [1 - P(b)] \tag{19}$$

If $P(a) = \frac{1}{2} + a$, and $P(b) = \frac{1}{2} + b$, then $P(c) = \frac{1}{2} + 2ab$. For equally likely inputs, the expectation is that $a = b = 0$, and the output is expected to be random. If *either* input is random, the output is expected to be random. If $a = b = \pm\frac{1}{2}$, $P(c) = 1$. If $|a| < \frac{1}{2}$, or if $|b| < \frac{1}{2}$, $P(c) = \frac{1}{2} \pm \min\{|a|, |b|\}$. This is the basic reason that the output from exclusive-or mixing of any number of inputs can be expected to enhance randomness characteristics. The advantage for UQS processing is that randomness is beneficial, at least on the average. A number of illustrative examples are derived in Ref. 11.

There is a down side to exclusive-or mixing: There are a large number of ways to get a correct output. In every exclusive-or mixing of q inputs, each containing 24 events, there are q^{24} ways to generate the "correct" output pattern. Many of these combinations have been scrutinized, but it isn't feasible to assess all combinations. The reasons that this multiplicity has been judged to be insignificant are that having only one way to generate a unique signal is an ideal that can never be assured in an abnormal environment, and there is no known threat that can utilize this effect to increase overall vulnerability.

4.2. One-Way Transforms

One-way transforms have been used in a number of weapon applications in order to derive information on whether or not intentional UQS entry has been accomplished successfully, without actually storing the correct UQS pattern. For these applications, the UQS pattern is temporarily captured in a buffer prior to driving the stronglink switch. Once used for a UQS buffer, erasure must be assured to a high level, commensurate with the assurance that a stronglink switch is reset (see Appendix C).

In an ideal UQS implementation, there would be no need for one-way transforms, because an echoed comparison with an inserted ROM key information source could be utilized[27]. It is also a deviation from ideal UQS implementation to use a buffer at all, but

[27] Jay Grear was the first to point this out to the author.

there are overarching operational considerations, such as the desire to deliver intent information prior to weapon release without actually driving a stronglink switch.

The general strategy of using a one-way transform is shown in Fig. 7. The human-initiated intent events (*r* bits) are transformed by the one-way transform into another *r* bits. A pre-determined one-way transform of the actual UQS pattern could stored outside the communication system (*e.g.*, in a log book) for comparison with the derived one-way transform, so that an indication can be provided back to the initiator of the intent as to whether the pattern was received correctly or not. The number of bits returned is intended to be *r*, but some systems have implemented a single-bit monitor return, which introduces a significant safety risk (discussed in Section 4.3).

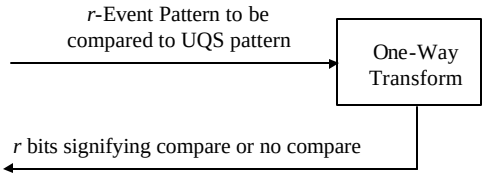

Figure 7. One-Way Transform Strategy

One-way transforms are intended to easily process the UQS pattern into a representation that is statistically different from the correct pattern, and such that the inverse transformation (producing the correct UQS pattern from the transformation) is extremely unlikely in normal and abnormal environments. Although not precisely defined, a one-way transform must be computationally straightforward to compute and computationally difficult to invert [Ref. 12].

The following notational description is summarized from Refs. 13 and 14. The transformation f has a domain X and a range Y such that every element of X is associated with a unique element of Y. One form of transformation is matrix multiplication[28], denoted in Eq. 20. First using notation for a general matrix:

$$[I_1, I_2, ..., I_k][M_{k\times p}] \oplus [X_1, X_2, ..., X_8] = [S_1, S_2, ..., S_p] \tag{20}$$

where a k-digit vector I is transformed to a p-digit vector S, and X is an optional initialization vector. For UQS applications, the association is one-to-one and onto, so that there can be no ambiguity about which transformation is associated with which input pattern. A one-to-one relation requires that the matrix be nonsingular (invertible) and therefore square (same number of rows and columns). For transformation of a modern UQS pattern, $k = p = 24$, and the matrix M is invertible. Therefore, the one-way properties must assure that is S is statistically different from I and M^{-1} cannot be easily derived (inadvertently) from M.

[28] Matrix transformation was used on the B83 and some modern safety upgrades, and would have been used on the W89 had it gone into production.

4.2.1 Initial Application of One-Way Transform

A "level one" transform (first phase of transform development), derived in the late 1970s, was based on the matrix structure of Eq. 20, but a matrix was chosen that was particularly amenable to microprocessor implementation. The mathematical description is:

$$[I_1, I_2, ..., I_8] \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \oplus [X_1, X_2, ..., X_8] = [S_1, S_2, ...S_8] \qquad (21)$$

where X is an initialization vector, $\oplus$ indicates GF(2) vector addition (bit-by-bit exclusive-or), and where the operation is repeated six times with the initialization vector used only the first time. The actual implementation utilized repeated shift-and-add operations, *i.e.*:

$$[S_1, S_2, ..., S_8] = [X_1, X_2, ..., X_8] \oplus [I_1, I_2, ..., I_8] \oplus 2 \times [I_1, I_2, ..., I_8] \bmod 2^8 \qquad (22)$$

where the modulo multiplication by two is implemented by a left shift without carry. The inverse operation, which is not designed into the system, is not implemented, and must be extremely unlikely to occur inadvertently in any environments, is:

$$([S_1, S_2, ..., S_8] \oplus [X_1, X_2, ..., X_8]) \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} = [I_1, I_2, ...I_8] \qquad (23)$$

The most straightforward implementation for the inverse operation that has been found[29] is:

[29] This expression is similar to one found in about the 1980 time frame by Curt Mueller.

$$[I_1, I_2, ..., I_8] = \sum_{i=1}^{8} 2^{i-1} \times ([S_1, S_2, ..., S_8] \oplus [X_1, X_2, ..., X_8]) \bmod 2^8 \quad (24)$$

The structure of Eq. 24 is similar to (although more complex than) Eq. 22. Since more dissimilar and difficult inversion is desirable, more sophisticated one-way transformations were developed in the 1990 time frame.

4.2.2. "Level Two" One-Way Transform

A more robust transformation was derived for each of the two 24-event modern double intent UQS patterns developed in the mid 1980s (Intent 1 and Intent 2); it is denoted in Eq. 25 [Ref. 15].

$$[S_1, S_2, ..., S_6] = [I_1, I_2, ...I_6] \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \oplus [X_1, X_2, ..., X_6] \quad (25)$$

The implementation was simplified according to Eq. 26, as represented by the software flow chart shown in Fig. 8.

$$S = X \oplus \sum_{i=1}^{6} I \times m_i \quad (26)$$

where the m_i are the six rows of the transformation matrix.

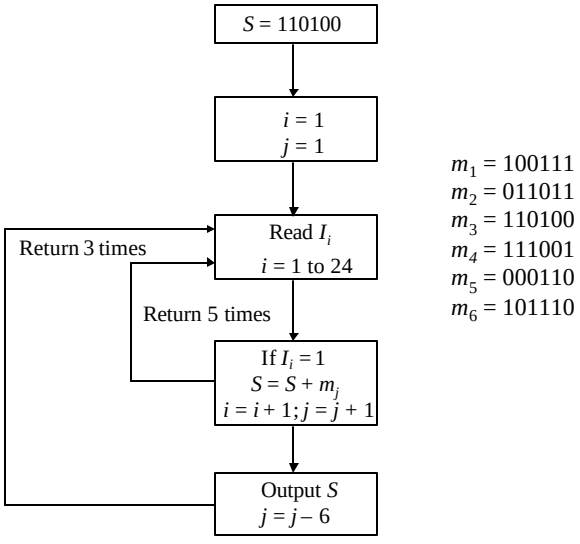

Figure 8. Software Implementation of Level Two Transform

The inverse transformation is:

$$([S_1, S_2, ..., S_6] \oplus [X_1, X_2, ..., X_6]) \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} = [I_1, I_2, ... I_6] \tag{27}$$

The most straightforward implementation known for the inverse transform (which is not implemented and would have to be inadvertently created) is similar to that for the forward transform, except that the six six-digit rows (matrix rows from Eq. 27) are significantly different than those of Eq. 25.

A polynomial multiplication transform representation was also developed in the late 1980s [Ref. 15]. The basic mathematical structure[30] is:

$$s(q) = i(q)\, p(q) (\text{mod } r^k) \tag{28}$$

where $p(q)$ is the transformation polynomial, and r is the base of the number system used. The polynomial notation is:

$$p(q) = a_{k-1} q^{k-1} + a_{k-2} q^{k-2} + ... + a_0 \tag{29}$$

where the information is contained in the a coefficients. Since r^k is not a prime number, the mathematical structure is a ring. Selected rings can give one-to-one transformations. In fact, a nonzero a_0 is necessary and sufficient for one-to-one transforms using this structure. Polynomial transforms of the type shown in Eq. 28 are equivalent to matrix transforms, and therefore inversion complexity is identical.

4.2.3. Exponential Transform

A class of exponentiation transforms was developed in the late 1980s. Known inversion techniques for these transforms are considerably more complex than for the matrix and polynomial transforms [Ref. 16]. The exponentiation is:

$$s = c^i \tag{30}$$

where c is a numeric constant and s and i are numbers or vectors. Exponentiation by an integer can be readily implemented by algorithms based on multiplication. The basic processor mode of multiplication modulos against the processor word length, n (modulo

[30] Polynomial transforms of this type can also be represented by matrices, but the notation is more complex.

2^n). This is a ring operation, but powers of two are frequently within one of a prime number, which would generate a field. For example, $2^4 + 1$ is a prime number, as are $2^8 + 1$ and $2^{16} + 1$. So are many others, such as $2^5 - 1$, $2^7 - 1$, $2^{13} - 1$, $2^{17} - 1$, $2^{19} - 1$, $2^{31} - 1$, and $2^{61} - 1$. These are not difficult to implement. The base number for the exponentiation must be primitive. For the applications described here, one half of the nonzero elements are primitive and can be found algorithmically [Ref. 15]. There are a variety of algorithms for efficiently computing exponentiation of these primitive numbers modulo a prime [Ref. 15] in order to obtain a one-to-one transformation.

One-way transforms generate as many bits of information as the number of events into the transform. The compression of this information into a smaller number of bits should not be done anywhere in the communication channel. The safety implications are discussed in Section 4.3.

4.3. Monitoring Unique Signal Pattern Correctness

The preceding section addressed the strategy of the one-way transform, including generation of 24 output bits for modern UQS patterns. An application producing the same number of monitor bits as input events was diagramed in Fig. 8. The purpose of this Subsection is to assess the use of n monitor bits, where $n < r$, as indicated in Fig. 9.

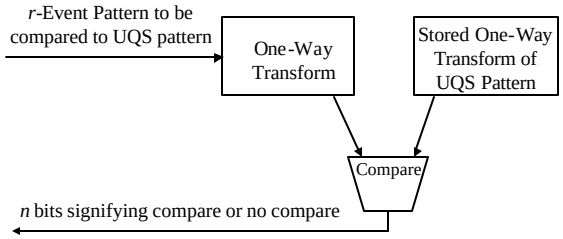

Figure 9. UQS Pattern Comparison Indication

The reduced number of monitor bits implies that since the recipient of the monitor bits is given fewer than 24 bits, there must be some pre-stored representation in the processing system of part of the correct one-way transform, and there is therefore some information about the correct UQS pattern contained in the n monitor bits. In the implementation shown in Fig. 10, $n = 1$ and a correct pattern is indicated by a logical "zero." Since multiple tries are possible, random inputs can be filtered to "trap" the correct UQS through an inadvertently (abnormal-environment) created "and" function.

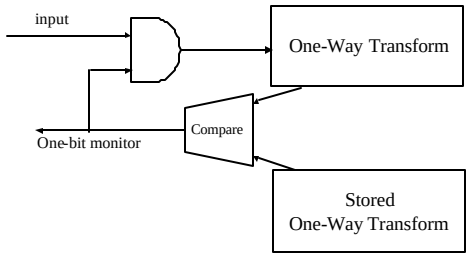

Figure 10. Trapping a Correct UQS Pattern

This is an illustrative concept; the trap can occur in a variety of ways (*e.g.*, "and" gate, shorted wires, software statements). The safety problem can be reduced (but not eliminated) by using additional monitor bits. For multiple monitor bits, the abnormal-environment-generated trap function is more complex in that it requires more inputs. For example it might require a separate software test for each monitor bit or a separate abnormal-environment-wired connection to the "and" gate. For multiple-bit monitoring, whether or not $n = r$, it is advantageous that there be a "pattern" to the monitor bits, *e.g.*, so that the trap logic is not allowed to be the same for all bits.

A quantitative metric and an analysis approach is addressed in this section for illustration. There should be no illusion about absolute quantitative accuracy; however, the comparisons are meaningful.

The model selected for this analysis is basically a "fault tree," shown in Fig. 11. Gate 1 shows the risk of inadvertently duplicating the unique signal, Gate 2 shows the risk of inadvertently trapping the unique signal, and Gate 3 shows the risk of inadvertently connecting the monitor bit (or pattern) to the trap function.

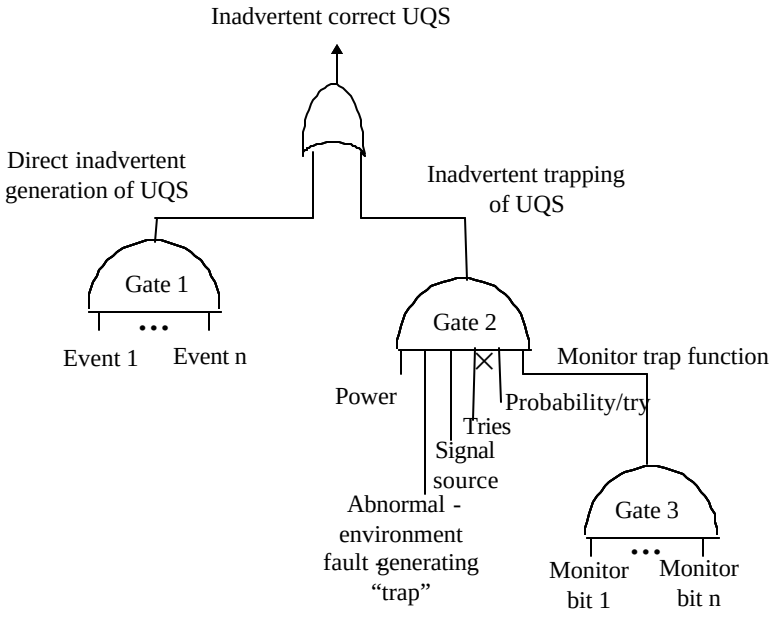

Figure 11. Fault Tree Model for Illustration of Monitor Problem

As a rough guide, an uncertainty range bounded by "P_{calc}" and a 1/3 logarithmic degradation[31] is helpful in illustrative assessments. For example, the Gate 1 output for a

[31] By incorporating various forms of dependence [Ref. 2], it has been found that an appropriate degradation limit is about $10^{\frac{2}{3}\log_{10} P_{calc}}$.

modern UQS pattern would be bounded by $\left(\frac{5}{11}\right)^5\left(\frac{6}{11}\right)^6\left(\frac{6}{12}\right)^6\left(\frac{6}{12}\right)^6 = 1.2\times10^{-7}$ and 2.5×10^{-5}.

The Gate 2 Power and Signal source inputs are generally considered to have a probability of one in an abnormal environment. The same assumption could be made for the existence of an abnormal-environment fault-generating trap. The number of tries and the probability per try are not actually "anded" or multiplied as indicated, but the logic is shown in this way for notational convenience. The probability of eventually generating the correct input (which means the correct monitor pattern) is actually:

$$P(pattern) = 1-(1-P(pattern/try))^{tries} \tag{31}$$

Gate 3 indicates the construction of the precise trapping function from the monitor bits, so it can be treated similarly to Gate 1. An example one-way transform monitor pattern of 48 events (*e.g.*, derived from double intent signals) is:

000000101011111010101010100011101000001010111011

For this, the P_{calc} result (with logarithmic degradation) is:

$$P_{calc} = \left(\frac{11}{25}\right)^{11}\left(\frac{14}{25}\right)^{14}\left(\frac{13}{22}\right)^{13}\left(\frac{9}{22}\right)^{9} = 1.2\times10^{-14} \text{ to } 5.3\times10^{-10} \tag{32}$$

For a 24-bit monitor pattern of:

000010101100110111000111

P_{calc} is (with logarithmic degradation):

$$P_{calc} = 1.2\times10^{-7} \text{ to } 2.5\times10^{-5} \tag{33}$$

For a single monitor bit, the range is about ½ to 2/3.

Assume that the inputs to the one-way transform were the C-module and D-module patterns. For this, P_{calc} is (with logarithmic degradation) 1.4×10^{-14} to 5.8×10^{-10}.

The results are shown in Table 6. The number of tries in a month for this calculation depends on how fast the one-way transform can be calculated. Based on (very rough) estimates made by component designers, 20 milliseconds is used here for the one-way transform computation time. From this, the number of tries per month for the one-way transform is 1.3×10^8. Note that uncertainty of the estimates is high.

Table 6. Computations Illustrating the Penalty of *n*-Bit Monitoring

	Gate 1	Gate 2	Gate 3	Result
One-way transform with one-bit monitor	1.4×10^{-14} to 5.8×10^{-10}	1.0	½ to 2/3	1.0
One-way transform with 24-bit monitor	1.4×10^{-14} to 5.8×10^{-10}	1.0	1.2×10^{-7} to 2.5×10^{-5}	1.0
One-way transform with 48-bit monitor	1.4×10^{-14} to 5.8×10^{-10}	1.6×10^{-6} to 1.4×10^{-4}	1.2×10^{-14} to 5.3×10^{-10}	1.6×10^{-6} to 1.4×10^{-4}

From these results, it would appear that there is a safety risk to using *n*-bit monitoring, where $n < r$. Although the risk can be reduced by adding more monitor bits[32], small numbers of additional bits aren't helpful for electronic transform speed[33].

[32] There are other methods possible, such as using incompatible communication (e.g., fiber optics) for the monitor function.

[33] The resettable MC 2969 has optional trapping contacts that can interrupt the input line upon successful enablement. It also has monitor contacts. However, it processes events at slow (mechanical) speed, which provides about a month before trapping can become a safety problem [see Ref. 1].

5. Future Work

Although the UQS concept has not changed significantly in the past three decades, UQS-related mathematical treatment can add to the understanding of potential response to abnormal environments. Another continuing objective is to apply newly developed mathematical tools to assure that problems are approached in as comprehensive a manner as possible. In this section, a summary list of potential future activities is provided.

1. The treatment of randomness metrics is extensive, but not complete. The methodology would benefit from a systematic exploration of the role of the collection in general and each specific metric in particular. The theory of complexity has never been sufficiently explored. The transformations of autocorrelation suggested by Ken Chen have not yet been pursued.
2. Joint randomness techniques for addressing the relations among sets of UQS patterns could benefit from the same type of developments.
3. One-way transforms are a concern because the known selection of inversion techniques will never be complete. This is analogous to certifying the robustness of encryption techniques without ever knowing all of the decryption techniques that might be possible.
4. The proof of Theorem Two (Appendix B) is not sufficiently general for mathematical comfort. It should be generalized beyond modern UQS patterns, and the margin should be certified. Jackie Leyland and a host of advisors attempted this work, but there was not sufficient mathematical background available at the time.
5. The assessment of signal-processing algorithms, such as described in Appendix A has depended on ad-hoc methodology pertaining to each algorithm as its implementation was attempted. Success depended on finding the right approach for unraveling each algorithm and finding it in time to call attention before the algorithm was committed. A more systematic mathematical approach would be useful.
6. The demonstration of increased variance due to dependent sources is compelling, but a comprehensive mathematical proof of the properties of deviations from random values would have value.
7. Possibilistic mathematical techniques appear to show more promise for analyzing uncertainty than do traditional probabilistic techniques. This is a relatively new field and should be followed for possible applications to UQS assessment methodology.

6. Conclusions

The UQS concept has withstood the test of time over the past thirty years. A variety of ill-advised attempts to change its features, with the aim of improving it, have proved futile, and many times counter-productive. Mathematical analysis has been one powerful means of gaining confidence in the inherent assurance of the concept. Although the mathematical aspects of unique signal assessment will never be complete, this documentation of the basis established to date should provide a useful future reference.

Appendix A: Subtle Dangers of Algorithmic Input Generation

The UQS methodology is straightforward, but unfortunately, many well meaning people have tried to implement schemes that deviate from a true UQS approach. Since these designs are usually much more complex than necessary for the UQS approach, they tend to mask problems with complexity. The complexity needs to be unraveled to strip away hidden safety problems. The approaches shown in this appendix aren't even safe in normal environments, but a major contributor to the shortfalls is that **none of the implementations described in this appendix were subject to abnormal-environment requirements**. The approaches described fall mainly into the areas of manual UQS event entry and trajectory-generated UQS events. The examples are offered generically, although they are all based on actual (non-Sandia) designs. They are not meant to precisely reproduce the designs or to disparage the designers, but rather are intended to demonstrate how carefully one must approach assessment of deviations from UQS methodology to identify subtle traps. This also helps to illustrate the importance of normal-environment scrutiny and abnormal-environment requirements.

A1. Masking Risk through Complexity

The following examples are intended to mathematically demonstrate that enhancing a principle-based approach may not enhance it at all; in fact degradation is possible (and perhaps likely).

A1.1. Masking Risk by Entry of 47 "Events" with Eight Keystrokes

UQS entry is intended to utilize serial data from a ROM key (or similar device that can be stored in an "inclusion region, separate from the communication system). Unfortunately, ill-advised use of keyboards is common. One example is a hexadecimal keyboard configured by putting the extra six keys directly above the standard decimal entry pad used on most computers and calculators. A keyboard layout is shown in Fig. 12, along with a logic circuit.

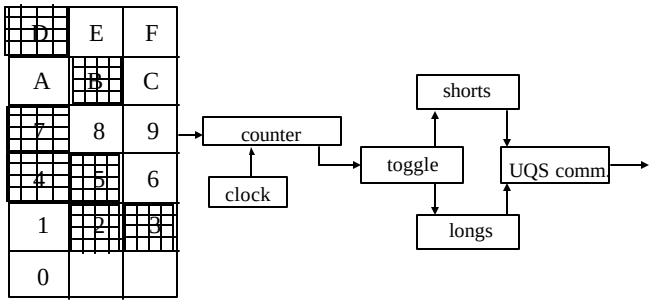

Figure 12. Ill-Advised Keyboard UQS Generator

Forty-seven separate, sequential bi-valued events are to be generated, but the design transforms these data entities into a compressed hexadecimal entry, B,2,D,3,5,2,7,4, using seven keys (shown shaded). Each keystroke generates a number that is loaded into a counter. Pulses are generated while the counter counts down to zero. The toggle assures transmission of first shorts, then longs, etc., so that the communication is 11 shorts, 2 longs, 13 shorts, 3 longs, 5 shorts, 2 longs, 7 shorts, and 4 longs. This is the MC

2969 pattern, but it is not how it was intended that it be generated. For example the third keystroke causes 13 "events" to be generated, whereas, the UQS events are intended to be separate and independent. The mathematical model that attempted to justify this design was based on equally likely independent hexadecimal entry:

$$P(UQS) = \left(\frac{1}{16}\right)^8 = 2.3\times10^{-10} \tag{34}$$

This is an example of a complex algorithm being represented by an oversimplified model. If only the seven actual keystrokes were considered[34], and if P(2|B) = P(7|2) = P(D|2) = P(3|D) = ½, and P(5|3) = P(2|5) = P(4|7) = ¾ (derived from Fig. 12), the result is:

$$P(UQS) = \left(\frac{1}{7}\right)\left(\frac{1}{2}\right)^4\left(\frac{3}{4}\right)^3 = 3.7\times10^{-3} \tag{35}$$

This result isn't an assured answer either, but it demonstrates that only slightly more complex assumptions (which don't look incompatible with the physical keyboard layout, given the Walske requirement) can change the result by seven orders of magnitude. In addition, groups of events in response to a single (keyboard) action violate the separate-event aspect of UQS resistance to statistically bias. This vulnerability to dependence is an important reason repeated selection from a set of two possibilities is a better safety approach than selection from a large set of possibilities where re-use of values is minimal.

A1.2. Masking Risk by Entry of 48 "Events" with Five Keystrokes

The design in this example was intended to generate a modern "double intent" UQS (two 24-event UQS patterns). The 48 events were compressed by the designers into a relatively simple keyboard entry, intended to be transformed into two 24-event UQS patterns. Fig. 13 shows a keyboard layout from which a mode of operation is selected (S), five keystrokes are entered (1,7,2,8,3), and "enter" is pressed (E). Note that there is no key re-use, which is an immediate danger flag. The digital processing is controlled by software, described here in words (but shown explicitly in Table 7). Five four-bit buffers are filled in response to the five information keystrokes.

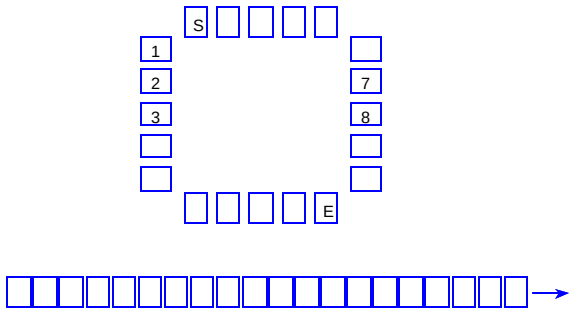

Figure 13. Keyboard Generation of Software Algorithm Input

[34] Perry D'Antonio asked the analysts who presented Eq. 34 if they thought surrounding the keyboard by many more unused keys might further enhance their perceived safety.

There is then a sequence of selections of four pointers, each selection of four made from five possibilities. The pointers select from the four-bit buffer contents. Each selection of four pointers causes retrieval of four of the five keystroke values. These 16 bits are added (exclusive-or) to one of a series of pre-stored values. This is done 96 times to generate 48 event words and associated CRCs (cyclic redundancy checks).

Table 7. Details on Ill-Advised Software Algorithm Example

```
Count=1
For A=0 to 10 Step 2
        For B=0 to 10 Step 2
                If B<>A Then
                        For C=0 to 10 Step 2
                                If C<>A and C<>B Then
                                        For D=0 to 10 Step 2
                                                If D<>A and D<>B and D<>C Then
                                                        W=Digit[A].SL.12
                                                        X=Digit[B].SL.8
                                                        Y=Digit[C].SL.4
                                                        Z=Digit [D]
                                                        FourDigin=W+X+Y+Z
                                                        XorConst=XORTable (Count*2)
                                                        Code=FourDigin XOR XorConst
                                                        Count=Count+1
                                                        If Count=96 Then Exit
                                                        EndIf
                                                EndIf
                                        Next D
                                        If D=10 Then Reset D=0
                                        EndIf
                                EndIf
                        Next C
                        If C=10 Then Reset C=0
                        EndIf
                EndIf
        Next B
        If B=10 Then Rest B=0
        EndIf
Next A
Exit
```

It is informative to dissect the 32 bits (information plus CRC) that are associated with each "event." The 16 CRC bits are for reliable transmission. Eleven of the remaining 16 bits are for communication overhead (type of communication, parity, odd/even toggle). Four of the five bits that define an "event" are effectively discarded to produce bi-valued events.

The result is that for each 12 cycles through the pointer selection process, only six bits determine how the UQS pattern will be generated. The six bit values depend only on whether the keystroke entered in one buffer (pointed to by the fourth pointer in each odd group of six pointers) is even or odd. The stronglink switch drive depends only on whether keystrokes are odd or even, not on which odd or even key is pressed. Therefore any entry sequence that reads odd, odd, even, even, odd (*e.g.*, 1, 1, 2, 2, 1) could drive both stronglink switches. There are 3125 such sequences out of the total population of 100,000. In addition, sharing keystrokes for two subsystems intended to be independent makes them highly dependent.

Another observation is that the design starts with the requirement to produce 48 bi-valued events (a potential population of 2.8×10^{14}). This is compressed into the information available in five keys selected from a population of 20 (potential population of 9.5×10^{13}). Through algorithmic manipulations, the input population is then reduced to an intermediate potential population of 3125, prior to re-expansion as it is then transformed back into 48 bi-valued events (potential population of 2.8×10^{14}). None of this was intentional; it was simply obscured by the inherent complexity of the algorithm.

A1.3. Masking Risk by Combinational Preparation of "Events"

The design described here was to "intent-enable" a trajectory stronglink switch (MC 2935) so that information from the trajectory flown by a missile would provide the additional information required to drive the stronglink switch. Unfortunately, there was no sequential characteristic to the signal processing until it was delivered to the stronglink switch.

The basic method was to send a combinational 32-bit seed word to the missile processor from which 16 dependent trajectory checks would each add (binary) to the accumulation on the seed word. The last 24 bits of the 32-bit sum were to correspond to the 24 events to the stronglink switch. In addition to compression of a requirement for 24 events into 16 trajectory checks, two of the checks were not flight-related.

The safety problems with the delivery of a combinational UQS pattern are exemplified in Section 3.3. Note that if a correct UQS pattern corresponded to a sector on a roulette-like wheel[35] containing $2^{24} - 1$ other sectors, there is no way to protect against statistically bias that might exist toward the sector containing the correct UQS pattern. However, if there were only two sectors, one containing an *A* and one containing a *B*, and the wheel were spun 24 times to select a pattern of 12 *A*s and 12 *B*s, the maximum threat would be for a balanced wheel, and bias due to sector preference would be eliminated.

Other problems in the design[36] are that the 16 trajectory checks are dependent in a potentially catastrophic manner. Each subsequent check pattern is derived from the previous pattern by circular right shifts, one shift for even-numbered checks and two shifts for odd-numbered checks. Aside from the dependence problem, if the first two checks were repeated eight times, the correct UQS pattern would be generated. Interestingly, the first two checks were preliminary to the missile flight.

A2. Masking Risk while Creating Bypasses

Deviating from the UQS methodology has always proved risky in the past. For example, there have been three proposals made by well meaning DoD contractors intended to compensate for DoD implementations that did not meet the UQS requirements. All three failed because of essentially the same oversight with respect to a "backdoor" unforeseen bypass operating in a different part of the implementation. The example depicted in Fig.

[35] This analogy was suggested to the author by Stan Spray in the early 1990s.
[36] Marty Fuentes was apparently the first to discover these (in the mid 1990s).

14 is a generic representation of the general strategy of three similar implementation proposals.

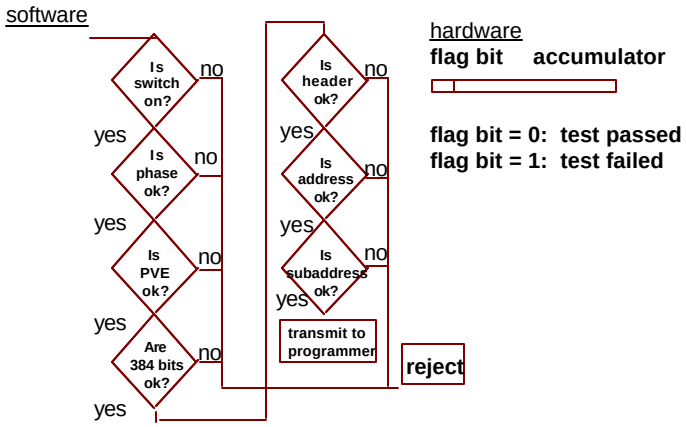

Figure 14. Ill-Advised "Enhancement" of UQS Implementation

Here, a variety of software tests (429 in the example) must be passed in order for any UQS communication to take place. For example, a hardware switch position must be correct. The computer processing must be in the correct phase. A communications handshake must be executed. Every bit of the 24 16-bit event representations must be correct. The UQS message header, the receiver address, and subaddress must be correct. If any one of these 429 tests fails, no UQS communication can take place. The expectation of the designers was that the probability of inadvertently passing the test where random results might be derived from each test[37] would be $2^{-429} \approx 7\times10^{-130}$.

This is a good example of focusing so hard on expected operation that backdoor bypasses are overlooked. The comparison for each test takes place by putting a value to be tested in a hardware accumulator, subtracting a reference value, and testing a flag bit for zero (test passed) or one (test failed), under potential abnormal-environment conditions. So a stuck flag bit can cause every test to be passed. These types of analyses also neglect the advantages of lockup features in preventing multiple tries (discussed in Section 4.3).

A mathematical exercise can help illustrate why supplementing the UQS methodology is fruitless. Consider a system that adds a large number of peripheral bits to the unique signal events and implements software discrimination to test the peripheral bits, which is in addition to the stronglink switch discrimination. A logic equation for safety failure (inadvertent acceptance of an input as the correct UQS) is:

[37] A sage piece of advise from Ref. 17: "--the naïve reliability calculator will often produce such absurd numbers as 10^{-16} or 10^{-18}. The low numbers simply say that the system is not going to fail by the ways considered, but instead is going to fail at a much higher probability in a way not considered."

$$
\begin{aligned}
P(\,failure) &= P(E_u)P(E_s)P(D_s)P(D_u) + P(E_u)P(\overline{E_s})P(\overline{D_s})P(D_u) \\
&+ P(\overline{E_u})P(E_s)P(D_s)P(\overline{D_u}) + P(\overline{E_u})P(\overline{E_s})P(\overline{D_s})P(\overline{D_u}) \\
&= [P(E_u)P(D_u) + P(\overline{E_u})P(\overline{D_u})][P(E_s)P(D_s) + P(\overline{E_s})P(\overline{D_s})] \\
&\approx [P(E_u)P(D_u) + P(\overline{E_u})P(\overline{D_u})]P(\overline{E_s})P(\overline{D_s})
\end{aligned}
\tag{36}
$$

where E_u indicates inadvertently correct events, E_s indicates inadvertently correct peripheral bits, D_s indicates successful software discrimination, D_u indicates successful UQS discrimination, and the over-bars indicate logical inversion. The first bracket depicts the modes of failure shown in Fig. 9. The approximation in the second brackets is made because in an abnormal environment, it is very difficult to give any assurance that software peripheral bits or discrimination will be correct. In fact, it would be difficult to give any assurance that the second bracket will have any value below one. This indicates that the addition of peripheral bits and software discrimination is insignificant compared to the safety offered by UQS generation and stronglink switch discrimination.

A3. Human vs. Environmental Inputs

Early ENDS systems depended entirely on trajectory-generated inputs for deriving the UQS for the trajectory stronglink switch. Without any human intent, the generation of trajectory unique signals is difficult to distinguish from an accident environment. One difficult requirement is that trajectory events should be independent. Designers have tried to introduce features such as "lanyard pull," "battery activation," "wing deployment," "motor firing," and sensing "S-turns," for example. Most of these lack independence; in fact they often occur in a prescribed sequence. Also, some designs have used two-sided tests to generate the correct unique event only if measured performance was between tightly constructed bounds (*e.g.*, in a particular computer phase, at a particular altitude, or traveling a particular speed). This strategy is counter-productive, because unexpected behavior can favor the *inverse* of the unique signal pattern, which is an inverter away from the correct pattern. The addition of human intent information to the trajectory safety subsystem was first introduced in the B77/B83 development in the late 1970s through "intent enablement" (mixing the intent signal with the trajectory-generated signal to drive the trajectory stronglink). This was only an interim solution, because the method compromised the requirement to keep the two safety subsystems as independent of each other as practical. Double intent was the ideal solution to the problem.

Sandia National Laboratories nuclear safety personnel have a long history of pushing for double intent, which requires two independent human-generated unique signals, preferably with ROM-key (or similar) sequential-event entry for both signals. Double intent was a basic component in the Sandia National Laboratories design for the Pershing II. The W89/SRAM II system and the W91/SRAM T, both of which, if fielded, would have been double intent from the warhead interface on. The Air Force agreed to increase the number of unique signal keystrokes used for SRAM II and provided two separate (but not independent) signals. Although far from an ideal source implementation, it would have been an important step in the right direction.

The best assurance of exceeding the Walske criterion of 10^{-6} in any credible abnormal environment is to have two independent safety subsystems that are each better than 10^{-3}. This is because the relation $10^{-3} \times 10^{-3} = 10^{-6}$ is not defensible unless the two subsystems are independent. Although intent enablement has been used in the past, and although the safety subsystems for the W89 and W91 would not have been independent on the DoD side of the WH interface, the nuclear safety goal for the safety subsystems has been to assure as much independence as practical. Intent and trajectory stronglink switches are both designed at Sandia National Laboratories and are both produced at the same production agency, but there is a concerted attempt to have the two stronglink switches be designed by different designers, operate on different principles, and respond differently to abnormal environment stimuli.

Appendix B: Subtle Problems in Driver Portion of Stronglink Switches

Stronglink switches translate electrical inputs to mechanical motion through "drivers" prior to discriminating UQS event types. This means that the driver methodology must support independence by assuring that no event response can depend on any other event response (prior to discrimination). Potential problems are discussed in this section.

B1. Dangers of Adding Dependence

Since events must be communicated independently, any dependence between how one event of a particular type is converted to drive in the stronglink switch to how any other event(s) of the same type is converted creates degradation in safety assurance. For example, it wouldn't be independent to signal an event of a particular type, say an *A*, by a two-volt increase in voltage so that the first *A* was communicated as an increase from zero to two volts, the second *A* was communicated as an increase from two volts to four volts, etc.

A less obvious dependence appeared in a stronglink switch design[38] in the 1990 time frame. Each odd event on an "*A*" input line was to be communicated by a positive change in voltage (low-to-high), and each even event was to be communicated by a negative change in voltage (high-to-low). The same process was to be used on a "*B*" line. Consider the C-module pattern, for which this stronglink switch was designed:

A,B,B,B,B,A,A,A,B,A,A,A,B,B,A,A,B,B,B,A,B,A,A,B

It can be shown mathematically that dependent drive signals can degrade the UQS concept. One example mathematical treatment is developed here. In order to establish a reference point, the resultant P_{calc} for first-order adjacent dependence is:

$$P_{calc} = \left(\frac{5}{11}\right)^5\left(\frac{6}{11}\right)^6\left(\frac{6}{12}\right)^6\left(\frac{6}{12}\right)^6 = 1.2\times 10^{-7} \qquad (37)$$

The dependent stronglink switch drive has four states, state *Q* (both lines low), state *R* (*B* line low, *A* line high), state *S* (*B* line high, *A* line low) and state *T* (both lines high). State changes would have to occur as:

R,T,R,T,R,Q,R,Q,S,T,S,T,R,T,S,T,R,T,R,Q,S,T,S,Q

Note that there are 8 transition pairs. The count for each is given in Table 7.

Table 8. Transition Counts for Dependence Example

Pair	*Q,R*	*Q,S*	*R,T*	*R,Q*	*T,R*	*T,S*	*S,T*	*S,Q*
Count	1	2	4	3	5	3	4	1

[38] The design was never used.

The maximum first-order adjacent dependence threat (using P_{calc}) for this sequence is:

$$P_{calc} = \left(\frac{1}{3}\right)\left(\frac{2}{3}\right)^2\left(\frac{4}{7}\right)^4\left(\frac{3}{7}\right)^3\left(\frac{5}{8}\right)^5\left(\frac{3}{8}\right)^3\left(\frac{1}{5}\right)\left(\frac{4}{5}\right)^4 = 5.1\times10^{-7} \quad (38)$$

The result indicates less safety than does P_{calc} for the C-module pattern, which is due to the dependence. In addition, the safety degradation (compared to an independent-drive design) becomes increasingly greater as higher-order dependence is considered. This is an undesirable ramification of non-independent stronglink switch drive.

B2. Advantages of Balancing Communication

As part of the interest in the driver portion of stronglink switches, safety assessment of stronglink switch design features that affect the balancing of event types is conducted. Although balancing is not crucial to the UQS concept, there are safety advantages pertaining to stronglink switch response . A subtle safety contribution to the safety of stronglink switches is afforded by a balanced response.

B2.1. The Driver Goal for Stronglink Switches

Using the UQS methodology, response to abnormal environments approaches that of random bi-valued sources, insofar as possible, even though threats can be nonrandom. This means that the maximum threat probability of inadvertent UQS duplication is forced by proper methodology to approach $\left(\frac{1}{2}\right)^{24}$ as closely as possible. The ½ would be met only if both types of events delivered from an abnormal environment were equally likely. The exponentiation would be valid only if all inadvertently generated events were independent of each other. But the threat environment must be assumed to exploit any vulnerability of the stronglink switch.

It might appear that balanced drive is less safe than imbalanced drive. Mathematically, if x is the probability of an A event type, and y is the probability of a B event type, $x^{12}y^{12} < \left(\frac{1}{2}\right)^{24}$ for $x + y = 1$ and $x \neq y$. However, an implementation that makes one event type very easy to generate runs the risk of discriminating strings of events of the same type in response to a single stimulus[39]. In order to illustrate how multiple-event response degrades safety, consider a correct trajectory pattern:

A,B,B,B,A,B,B,A,A,A,A,B,B,A,B,A,B,A,A,B,B,B,A,A

where strings of multiple *B*s could be discriminated in response to each individual *B* stimulus. In this case, the correct pattern could be generated by the input:

A,B,A,B,A,A,A,A,B,A,B,A,B,A,A,B,A,A

[39] This problem is examined in more detail in Appendix B3.

if the extra *B*s in a string were skipped. This could result in a loss of 6 events' assurance if strings of *B*s were discriminated in response to a single stimulus.

There are other considerations, such as the degree of imbalance and the degree of resistance to abnormal-environment skipping. However, the safest goal appears to be to design a balanced stronglink switch response to event types.

B2.2. Imbalance Features in Stronglink Switch Design

One example of imbalance was found[40] in a conceptual stronglink switch design, which used a "gate/drive" strategy, meaning that there were separate inputs for moving a gate arm and for driving the stronglink switch to its next event response position. The stronglink switch design had two types of imbalance.

One type of imbalance was mechanical. The gate arm in the conceptual design was spring loaded so that an *A* event type required energizing a gate solenoid to move the gate arm in opposition to the spring while the drive was energized. The response for a *B* event type required no gate solenoid energy during the drive, which made *B*s easier to generate than *A*s, because the drive action corresponding to *A*s required opposition to the spring and the drive action corresponding to *B*s required no spring action.

The other type of imbalance was electrical. For an *A* event in the conceptual design, information would have had to be available on both the gate and drive lines in order to energize the gate solenoid and provide drive. For a *B* event, information need only be available on the drive line.[41]

B3. Multiple-Event Response to Single Inputs

The UQS approach is predicated on the stronglink switch either accepting one type of event or the other, or not responding at all (which would be a "non-event"). Stronglink designers have had to address a deviation from this behavior in various stronglink switch designs, which is known as "multiple-event response" (runs) to a single information input. In some designs, a particular environmental stress is required in combination with the input. In some designs, multiple-event response is limited to advances only of positions having the same type as the initiating event, and in others there is unrestricted advance. Newer stronglink designs have incorporated features such as anti-run "tabs."

However, there has been some contention from time to time that stronglink switches that have run tendencies are not only "safer" than single-event-single-response (run-free) stronglink switches, but that random response is the upper bound on the probability of an inadvertent correct response. This is false, as the Theorems One and Two[42] demonstrate.

---

[40] This was found and identified as a potential problem by Ken Eras, who is in Rich Kreutzfeld's stronglink design organization.

[41] A similar imbalance existed in the MSAD, a stronglink switch where an *A* is communicated by a mechanical "pull," and a *B* is communicated by a mechanical "pull," followed by a mechanical "push."

[42] The basics of this analysis were first derived in late 1994 and early 1995.

There are 2^{24} possible patterns of twenty-four bi-valued events. Modern UQS patterns have 12 runs of length one to four. Since $4^{12} = 2^{24}$, it has been suggested by some that there is no difference between 12 multiple-event responses where there are four possible run lengths and 24 single-event responses. Theorem One demonstrates that 12 selections from four possibilities results in more vulnerability to first-order-adjacent dependence than does 24 selections from two possibilities, assuming balanced events and balanced transitions.

Theorem One: Making 12 sequential choices from among four run-length possibilities, for a total of 24 entities increases maximum vulnerability to inadvertently matching balanced (event-wise and transition-wise) patterns, when compared to making 24 sequential choices from two possible event types, whether or not the choices are independent.

Proof: $x(q)$ signifies a process of making r selections from a set having s members. The process is applied to an event-wise and transition-wise balanced pattern, s, which has $r_s = 24$, and $s_s = 2$, and 12 *A*s and 12 *B*s. The process is also applied to a "run" pattern, t, corresponding to groups derived from an event-wise and transition-wise balanced pattern, which has $r_t = 12$, and $s_t = 4$, and w run lengths r_1, (length one), x of r_2, (length two), y of r_3 (length 3), and z of r_4 (length four). Note that $w + 2x + 3y + 4z = 24$

First, consider the process t. $P(r_1) + P(r_2) + P(r_3) + P(r_4) = 1$. A Lagrange solution [*e.g.*, Ref. 18] is:

$$P(x(q) = t) + Lg = P(r_1)^w P(r_2)^x P(r_3)^y P(r_4)^z + L[P(r_1) + P(r_2) + P(r_3) + P(r_4) - 1] \quad (39)$$

where L is the Lagrange multiplier, g is the side condition constraining the sum of the probabilities to one, and w, x, y, and z are the number of appearances of r_1, r_2, r_3, and r_4, respectively. Taking partial derivatives, four equations are obtained, accompanying the side condition:

$$wP(r_1)^{w-1} P(r_2)^x P(r_3)^y P(r_4)^z + L = 0 \quad (40)$$

$$xP(r_1)^w P(r_2)^{x-1} P(r_3)^y P(r_4)^z + L = 0 \quad (41)$$

$$yP(r_1)^w P(r_2)^x P(r_3)^{y-1} P(r_4)^z + L = 0 \quad (42)$$

$$zP(r_1)^w P(r_2)^x P(r_3)^y P(r_4)^{z-1} + L = 0 \quad (43)$$

$$P(r_1) + P(r_2) + P(r_3) + P(r_4) = 1 \quad (44)$$

Multiplying Eq. 40 by $P(r_1)$, Eq. 41 by $P(r_2)$, Eq. 42 by $P(r_3)$, and Eq. 43 by $P(r_4)$, and adding, the solution for L is derived, from which the four maximum threats can be obtained:

$$(w + x + y + z)P(r_1)^w P(r_2)^x P(r_3)^y P(r_4)^z + [P(r_1) + P(r_2) + P(r_3) + P(r_4)]L = 0 \quad (45)$$

$$L = -(w + x + y + z)P(r_1)^w P(r_2)^x P(r_3)^y P(r_4)^z \tag{46}$$

The greatest threat is therefore:

$$\begin{aligned} P(r_1)_{\max} &= \frac{w}{w+x+y+z} \\ P(r_2)_{\max} &= \frac{x}{w+x+y+z} \\ P(r_3)_{\max} &= \frac{y}{w+x+y+z} \\ P(r_4)_{\max} &= \frac{z}{w+x+y+z} \end{aligned} \tag{47}$$

where:

$$P(UQS)_{\max} = \frac{w^w x^x y^y z^z}{(w+x+y+z)^{w+x+y+z}} \tag{48}$$

From this, the maximum resistance to arbitrary threats would be to choose each of the variables according to:

$$w + x + y + z = 4w = 4x = 4y = 4z \tag{49}$$

Since $w + x + y + z = 12$, this would require $w = x = y = z = 3$.

However, *w*, *x*, *y*, and *z*, cannot be made equal, because of the constraint on the number of events, which requires that $w + 2x + 3y + 4z = 24$. Let $2^{-24} = a$, a bound which would require that *w*, *x*, *y*, and *z* be equal. Since *w*, *x*, *y*, and *z* must be non-equal, then $P(x(q) = t) \neq a$, and since a is a minimum bound, $P(x(q) = t) > a$.

Next, consider the process s. $P(A) + P(B) = 1$, where there are *a A*s and *b B*s..

$$P(x(q) = s) + Lg = P(A)^a P(B)^b + L[P(A) + P(B) - 1] \tag{50}$$

Following an approach similar to the preceding:

$$\begin{aligned} &aP(A)^{a-1} P(B)^b + L = 0 \\ &bP(A)^a P(B)^{b-1} + L = 0 \\ &P(A) + P(B) = 1 \\ &(a+b)P(A)^a P(B)^b + [P(A) + P(B)]L = 0 \\ &L = -(a+b)P(A)^a P(B)^b \\ &P(A)_{\max} = \frac{a}{a+b} \\ &P(B)_{\max} = \frac{b}{a+b} \\ &a = b = 12 \end{aligned} \tag{51}$$

Therefore, the maximum vulnerability for event-wise and transition-wise balanced patterns can be constrained to $P[x(q) = s] = a$. This proves Theorem One for the case where threat selections are made independently. Dependent selections (e.g., first-order adjacent dependence) give similar results. The optimum first-order-adjacent-dependence-resistant pattern (balanced events and balanced transitions) for 24 bi-valued events gives a first-order-adjacent-dependence-degraded form of a, a bound which is termed a^*:

$$P(UQS)_{\max} = \left(\frac{6}{12}\right)^6 \left(\frac{6}{12}\right)^6 \left(\frac{6}{11}\right)^6 \left(\frac{5}{11}\right)^5 = 1.2\times10^{-7} = a^*$$

Even if the run transitions could be perfectly balanced, the result for four run lengths (r_1, r_2, r_3, and r_4, would be greater than a^*, which is sufficient for the proof.

$$P(UQS)_{\max} = \left(\frac{1}{4}\right)^8 \left(\frac{1}{3}\right)^3 = 5.7\times10^{-7}$$

But the run patterns and transitions cannot be balanced due to the constraint on the number of events. The optimum first-order-dependence-resistant 12-run sequence that is analogous to 24 events (optimum balanced events, optimum runs satisfying the constraints requiring 12 runs and 24 events, and balanced run transitions) results in:

$$P(UQS) = \left(\frac{2}{5}\right)^2 \left(\frac{1}{5}\right)^3 \left(\frac{1}{2}\right)^2 \left(\frac{1}{2}\right)^2 \left(\frac{1}{2}\right)^2 = 5\times10^{-6} > a^*$$

Examples:

For the D-module pattern[43], s = *A,B,A,A,A,A,B,A,A,B,A,A,B,B,B,B,A,B,B,B,B,A,A,B*. If $P(A) = P(B) = ½$, $P[x(q) = s] = a = 5.96\times10^{-8}$. The P_{calc} result for first-order-adjacent dependence is 1.2×10^{-7}, which equals a^*.

For the "runs" equivalent pattern, $t = r_1r_1r_4r_1r_2r_1r_2r_4r_1r_4r_2r_1$, where the runs begin with *A* and alternate between *A* and *B*. If $P(r_1) = ½$, $P(r_2) = P(r_4) = ¼$, and $P(r_3) = 0$, $P[x(q) = t] = 3.8\times10^{-6} > a$. The P_{calc} result for first-order adjacent dependence is 7×10^{-4}, which is greater than a^*.

Theorem Two demonstrates that the existence of runs (multiple discriminated events in response to single stimuli) can increase threat vulnerability over that of run-free discrimination.

[43] The same result is obtained for any modern UQS pattern.

Theorem Two[44]*:*

Independent multiple-event responses to balanced (event-wise and transition-wise) stimulus patterns can have a maximum inadvertent $P(UQS)$ that exceeds a, whereas independent run-free responses cannot.

Proof:

This proof is through demonstration of a condition for exceeding a, based solely on independent run likelihood. Independent run-free response $P(UQS)_{\max} = a$ for any balanced pattern.

Express the D-module UQS pattern as $r_{A1}r_{B1}R_{A4}r_{B1}R_{A2}r_{B1}R_{A2}R_{B4}r_{A1}R_{B4}R_{A2}r_{B1}$, where the runs are denoted as runs of *A*s and runs of *B*s. Since the capitalized runs can be constructed from sub-runs, the potential constituents are:

$$R_{A4} = r_{A4} \cup {r_{A2}}^2 \cup 2r_{A3}r_{A1} \cup 3r_{A2}{r_{A1}}^2 \cup {r_{A1}}^4 \qquad (52)$$

$$R_{A2} = r_{A2} \cup {r_{A1}}^2$$

$$R_{B4} = r_{B4} \cup {r_{B2}}^2 \cup 2r_{B3}r_{B1} \cup 3r_{B2}{r_{B1}}^2 \cup {r_{B1}}^4 \qquad (53)$$

$$R_{B2} = r_{B2} \cup {r_{B1}}^2$$

where $\cup$ indicates logical "or" and juxtaposition indicates logical "and."

Substituting $P(r_{A4}) = P(r_{A3}) = P(r_{B3}) = P(r_{A2}) = P(r_{B2}) = 0$, $P(r_{A1}) = \frac{1}{2}$, and $P(r_{B1}) = \frac{1}{2} - P(r_{B4})$, converts the probability equation to:

$$P(r_{A1})^{12} P(r_{B1})^4 (P(r_{B4}) + P(r_{B1})^4)^2 = a\,(1-2P(r_{B4}))^4 [16P(r_{B4}) + (1-2P(r_{B4}))^4]^2 \qquad (54)$$

The expression multiplying a is: $1 + 8P(r_{B4}) + 8\,P(r_{B4})^2 - \ldots$ (13 terms). For any small values of $P(r_{B4})$, the expression is greater than one, so $P(UQS) > a$. This proves Theorem Two.

Example:

Consider the C-module pattern[45]. Let $P(r_{B4}) = 0.11$, $P(r_{A1}) = 0.66$, and $P(r_{B1}) = 0.23$. The result is $P(UQS) = 2.43\times10^{-7}$ (which is more than four times greater than a).

[44] Dick Schwoebel, then SNL Director of Surety Assessment, not only encouraged the development of the proof, but also did his own empirical calculations in support.

[45] Similar results are also obtained for the D-module pattern.

B4. Partial Event Generation

Drivers that respond to a UQS event input are intended to cause discrimination relatively quickly or to remain in or return to a quiescent state. Drivers that move progressively over a period of time and do not return to a quiescent state have created safety concerns from a unique signal (UQS) safety standpoint[46]. The concerns fall into two categories: 1) multiple-event processing, and 2) potential ambiguity of event types.

B4.1. Multiple-Event Processing

A stronglink switch meeting the UQS concept [Ref. 1] should respond to one communicated event by moving one discriminator position. In normal and abnormal environments, multiple-event discriminator movement during shock or electrical stress should be precluded, and in normal environments, multiple-event discriminator processing in response to a single communication of information should be precluded. Both of these constraints are difficult to meet for progressive drivers, because the progression depends on how long the signal is applied. For example, a signal of less than the intended duration can drive part way toward discrimination of an event of a particular type, and a signal of greater than the intended duration can drive through discrimination of multiple events of a particular type. For any safety subsystem, the UQS is intended to be a 24-event pattern that is the only sequence that can cause the switch to reach the pre-arm position. Since UQS patterns are meticulously engineered for uncertainty and resistance to dependence, any other pattern inadvertently accepted by a switch could be vulnerable to common characteristics that threats can have. Also, acceptance of multiple patterns approximately sums the probability of inadvertently receiving each pattern.

The most commonly used assessment tools help illuminate the problems. First, the UQS concept requires that 24 separate bi-valued communications. Although 2^{-24} is 5.96×10^{-8}, 24 events are considered necessary for use in a 10^{-3} safety subsystem for a variety of reasons [Ref. 1]. Therefore, anything that affects this level of safety significantly is a safety concern. One of the tests is for balance of events having each communication type (*e.g.*, equal numbers of *A*s and *B*s). The event-type-balance test for the pattern communicated to the stronglink switch drivers[47] has 12 *A*s and 12 *B*s, resulting in an event-type-balance test value of 5.96×10^{-8}. A second fundamental test is called a "first-order adjacent dependence" or P_{calc} test [Ref. 1], which basically allows the threat to furnish an event in response to a previous event with the same probability as the ratio of transition pairs in the UQS pattern. For a pattern such as that to which a piezoelectric driver might be designed to respond (with 6 *AA*s, 6 *AB*s, 5 *BA*s, and 6 *BB*s), the maximum threat is 1.2×10^{-7} [Ref. 1].

However, the intended pattern is only one of an extremely large number of patterns to which progressive drivers can respond, in normal or abnormal environments. For example, assume that an event drive is designed to be duration T. If the stronglink switch

[46] Piezoelectric drivers that have been proposed to date for stronglink switch applications have these undesirable characteristics.

[47] The pattern considered is *A,B,B,B,B,A,A,A,B,A,A,A,B,B,A,A,B,B,B,A,B,A,A,B*.

is ready for the first event in the pattern, and if there is a drive of duration T on the *A* line, the event will be discriminated as correct. Next, if there is a drive of duration 4T on the *B* line, the next four events will be discriminated as correct. In fact, it only takes 12 signals, applied alternately to the *A* and *B* lines to drive the discriminator to the pre-armed position. Now look at this as a potential threat pattern.

Suppose the lengths of the drive signals in the threat pattern are classified into T, 2T, 3T, and 4T. The drive required to prearm a progressive-driven stronglink with drive alternately applied to the *A* and *B* lines, starting with the *A* line is T, 4T, 3T, T, 3T, 2T, 2T, 3T, T, T, 2T, T. Under the alternating drive assumption, the number of ways this can be done is 4^{12} (the population of 12 selections from among 4 choices), which is the same as 2^{24}, so there is no degradation under the first basic test. However, in the threat pattern, there are five Ts, three 2Ts, three 3Ts, and one 4T. Suppose the threat has the same imbalance that the pattern has (as is a normal test for any UQS pattern). The maximum threat is then $(5/12)^5(3/12)^3(3/12)^3(1/12) = 2.5\text{x}10^{-7}$. This is about a factor of four worse than a balanced signal. Now apply the first-order adjacent dependence test. The transition pairs are: T followed by 4T once, by 3T once, by 2T once, and by T once; 2T followed by T once, by 2T once, and by 3T once; 3T followed by T twice, and by 2T once; and 4T followed by 3T once. Allowing the threat to have the same frequency of occurrence, we get $(1/4)^4(2/3)^2(1/3)^4 = 2.1\text{x}10^{-5}$. This is about a factor of 200 worse than a balanced pattern. Tabulating:

Table 9. Testing a UQS Threat for Progressive Drivers

	Balanced Signal	Threat Signal
Population Test	$5.96\text{x}10^{-8}$	$5.96\text{x}10^{-8}$
Type Balance Test	$5.96\text{x}10^{-8}$	$2.5\text{x}10^{-7}$.
Transition Balance Test	$1.2\text{x}10^{-7}$	$2.1\text{x}10^{-5}$

The progressive driver fails two of the most important tests that can be applied.

B4.2. Ambiguous-Event Response

In the UQS concept, it was intended that every input to the stronglink switch would be recognized as an *A*, as a *B*, or not recognized as either, which would be a "non-event" [Ref. 1]. There is another situation possible for progressive drivers. Suppose the drive has been part way toward an event (input T/*n*, where $n > 1$). If there is a position such that one event (say an *A*) has not been discriminated, but the opposite-type event (*B*) cannot be accepted as an input (by drive of duration T on the *B* line), then what has occurred is not an *A*, nor a *B*, nor a non-event. Therefore, it compromises the assurance that would be possible if the UQS concept were followed.

Appendix C: Problems Due to Testing/Exercising

C1. Testing/Exercising

It is essential to the safety of UQS systems that the UQS information not be intentionally stored, especially where it can be directly used to drive a stronglink switch. A problem arises in testing or exercising the system if the actual UQS pattern is intentionally stored in the communication system, because it might not be erased, in spite of a desire and requirements to do so. There is a similar concern for component testing. One of the reasons for this is the variety of ways in which human or hardware error can lead to remnant information. Another reason is the difficulty in preventing memory bias toward the last data or the most common data entered (even in "volatile" memories). Some of these problems are indicated in this section through a fault tree description.

C2. Fault Tree Illustration

The assumptions used in the fault tree were that the correct UQS was inserted into the communication system during testing or exercising, and that memory favors the most commonly used information inserted, which is a pervasive memory problem. Note that none of the fault tree threat space is present if the correct UQS is not entered.

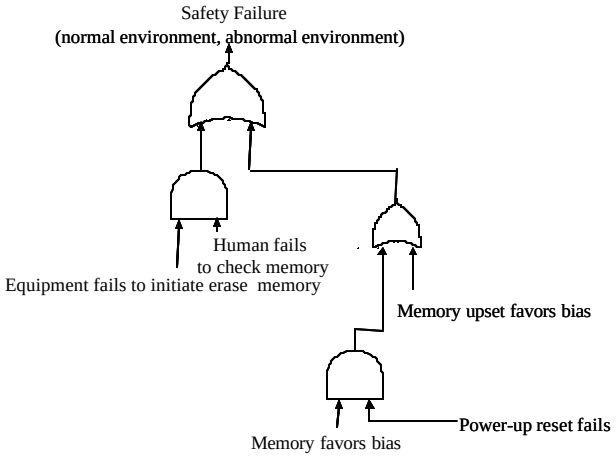

Figure 15. Test/Exercise Illustrative Fault Tree

The fault tree illustrated has two main paths:

1. A normal-environment path for the safety threat that the correct UQS information is not erased from non-volatile memory
2. An abnormal-environment path (with some normal-environment threat) for volatile or non-volatile memory changing to the information most commonly inserted

The first path can occur if the test equipment fails to erase memory and this failure is not detected by a human.

The second path addresses the potential for a memory prejudice during start-up that is not over-ridden by a memory reset routine, or for a memory "upset" due to a power glitch or radiation burst.

At a component level, assurance that stronglink switches are delivered in a reset condition is given by two independent tests; one electrical, and one through radiographic examination.

A remedy for the system test problem is to test only with "safe patterns" [Ref. 1], which are significantly different from UQS patterns and allow testing every functional part without ever using a UQS pattern.

Appendix D: Mathematical Descriptions of Uncertainty[48]

If kept in perspective, mathematics can be useful in addressing uncertainty with respect to meeting quantitative requirements and with respect to comparing implementation approaches. The perspective necessary is that mathematical descriptions depend on modeling choices, and these can easily differ from reality, especially in abnormal environments. With this perspective and these cautions as background, it is informative to consider various mathematical approaches. First, consider a probability density function (PDF) as a possible tool for portraying uncertainty[49].

D1. The Gaussian ("Normal") Distribution

A particular PDF is shown in Fig. 16 (a normalized form of a Gaussian or "Normal" distribution).

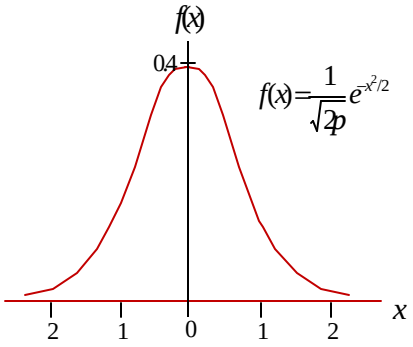

Figure 16. Normalized Gaussian Distribution

The Central Limit Theorem [*e.g.*, Ref. 4] establishes conditions for which the sum of a large number of independent but not necessarily identically distributed random variables, where none of the variables is dominant, is expected to be approximately Gaussian. This might explain why this PDF is sometimes used to represent uncertainty about complex processes. For example, the abscissa has been used to portray uncertainty about the time before a component might wear out [Ref. 17]. But there are restrictive assumptions in the Central Limit Theorem proof. For example, the application to independent processes and large numbers of non-dominant processes constrain threats to a small subset of possibilities. A more subtle constraint is that the summation abscissa is assumed linear (*e.g.*, logarithmic processes must be treated separately). Another problem is that the abscissa limits for nonzero ordinate values are unlimited in extent. This means that parameters approaching infinity are not precluded and uncertainty about probabilities requires truncated forms of the distribution. None of this rules out the Gaussian distribution (or any similar distribution) as an approximate representation of uncertainty.

[48] "Uncertainty" means lack of precise knowledge, which is distinguished from "variability," for which variation likelihoods are known precisely.

[49] Probability density functions are actually only appropriate for depicting variability (variation due to processes that can be physically modeled), but illustrative uses are not precluded if kept in perspective.

But because of the limitations above, overconfidence in results derived from this starting point is a potential danger, as will be illustrated in Section (D2).

D2. The Uniform Distribution

In order to represent uncertainty between two limits, a uniform PDF is sometimes used, occasionally citing Pierre-Simon Laplace (French mathematician) as assuring that lack of knowledge corresponds to equal likelihood for any value across an interval, or citing Claude E. Shannon as "proving" that uniform distributions represent maximum entropy (minimum information) [Ref. 7].

It is doubtful either Laplace or Shannon thought equal likelihood could be used in probabilistic safety analysis to represent lack of information, especially since a PDF exactly specifies likelihood. An example demonstrating the analytical penalty of over-specifying knowledge is to consider the probability of throwing two heads with two independently thrown deformed coins whose deformation characteristics are unknown. If the probabilities of throwing heads, $P(h_1)$ and $P(h_2)$, are unknown, with densities $f_1(x)$ and $f_2(x)$, the probability density of throwing two heads for independent trials would be given by [Ref. 18]:

$$P(h_1) \cap P(h_2) = f(y) = \int_0^\infty \frac{f_1(x)\, f_2(y/x)dx}{x} \tag{55}$$

where $\cap$ signifies conjunction (logical "and"). When $f_1(x)$ and $f_2(x)$ are uniformly distributed over [0, 1], the result is shown in Fig. 17.

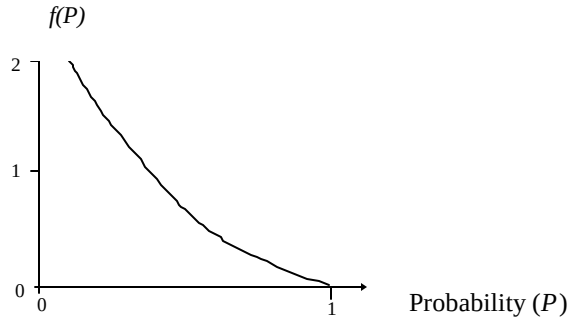

Figure 17. PDF for Conjunction of Two Independent Uniform Distributions

Four of many possible concerns about this "provable" result are: 1) the conjunction operation assumes independent deformation, where no such assurance was given, 2) the mean probability of throwing two heads is derived from this result as exactly ¼ (no uncertainty), 3) the right-most extreme result (probability one of throwing two heads) appears vanishingly small, even though there was no information given that would warrant that result[50], and 4) since there is no more information about the probability of

[50] In fact, deformations that rolled the coins into cylinder-like shapes could easily make the probability of two heads exactly equal to one.

throwing two heads than there is about the probability of throwing one head, maximum entropy for the conjunction could also be assumed to be a uniform distribution. The problem with the maximum entropy approach is that assuming information beyond that given can make "proofs" inapplicable to assured safety analysis modeling. Since these results readily extend to calculating the probability of throwing 24 heads with 24 deformed coins, the main lesson for unique signal analysis is that there is no assurance that the probability of inadvertently receiving a correct unique signal pattern in an abnormal environment is $\left(\frac{1}{2}\right)^{24}$, even as an average, much less as an assurance, given any credible exposure. This was demonstrated in Section 3 of this report.

As an illustration, the PDF for the conjunction of 24 inadvertently correct abnormal-environment-generated independent events (each modeled by a uniform distribution) would be as shown in Fig. 18, where the "delta function" symbol represents a *very* narrow PDF in the vicinity of 6×10^{-8}.

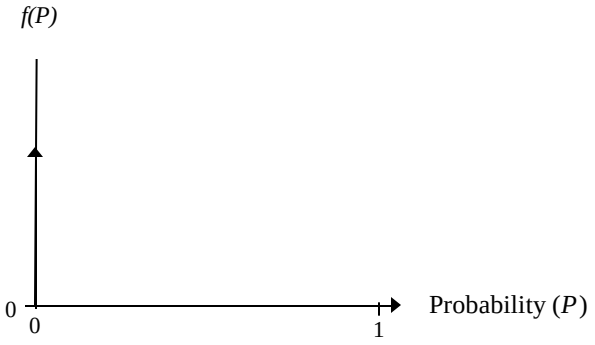

Figure 18. PDF for the Conjunction of 24 Independent Uniform Distributions

Approximately the same result is obtained if truncated Gaussian distributions are used. These types of analyses can give the impression that a 24-event UQS is nearly impossible to inadvertently duplicate. As was demonstrated in Section 2.4, this can be overly optimistic, because of dependence in the generation process and modeling uncertainty. Because it is easier for inadvertent processes to inadvertently duplicate 24 correct events than the above analyses indicate, there are special features of UQS methodology that are a necessity. These include carefully engineered pattern design (engineered "uncertainty"), carefully controlled separate event communication (no packaged or numbered communications), and precautions against leaving memory traces of the correct UQS (due to human mistakes or equipment malfunctions). All of these are addressed in this report.

First, some approaches that are not based on probability distributions will be reviewed.

D3. Fuzzy/Possibilistic/Hybrid Descriptions of Uncertainty

Possibilistic numbers represent knowledge uncertainty without injecting more information than is available. Fuzzy numbers are a special case of possibilistic numbers.

Interval numbers are a special case of fuzzy numbers [Ref. 19]. Hybrid numbers have both a probabilistic part and a fuzzy part [Ref. 20]. Probability-Bound numbers show the range of probability distributions possible for a given amount of information [Ref. 21]. All of these can help give insight into UQS safety likelihood, although none offer definitive solutions. A sample of some of the advantages of multiple views through multiple mathematical models follows.

D4. Uncertainty about 24 Abnormal-Environment-Generated Events

Fuzzy and possibilistic representations of completely unknown probabilities of inadvertently correct abnormal-environment-generated events look like uniform distributions (constant ordinate value across the abscissa range from 0 to 1), but mean something much different. The meaning for possibilistic numbers corresponds exactly to the available knowledge (there is no representation about the likelihood of unknown regions), and no assumed information is injected as a consequence of the uncertainty model. The conjunction of the possibilistic functions for inadvertently generating 24 correct events (an inadvertently correct UQS pattern) is shown in Fig. 19. Note the contrast with Fig. 18.

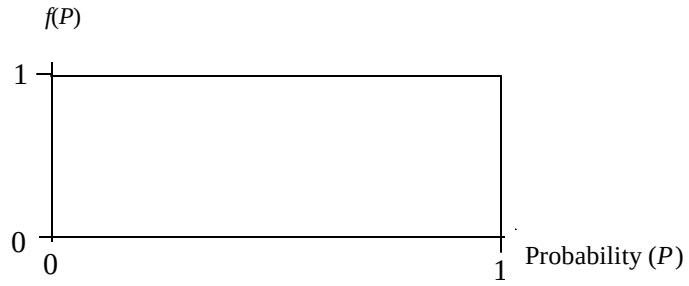

Figure 19. Conjunction of 24 Unknown Independent Fuzzy Events

This same result is obtained for interval, fuzzy, possibilistic, and Probability-Bounds numbers. At this point, the PDF approach appears overly optimistic with respect to safety, and the other approaches seem incapable of giving any useful information. However, each approach can find utility in a different way. For example, in some situations, measurable physical response data could improve the PDF approach as well as the Probability-Bounds and hybrid approaches. Formally elicitated expert engineering judgment could improve the interval, fuzzy, possibilistic, hybrid, and Probability-Bounds approaches. None of this helps address quantitative safety requirements, but the insights into uncertainty estimates that can be obtained are important.

As an example, consider inadvertently generated events, each having uncertainty represented by a triangular probability distribution with mean value of one-half and extremes ranging from probability zero to probability one, compared to each having uncertainty represented by a triangular fuzzy function ranging from probability zero to probability one with a peak at one-half (Figure 20). The conjunction of 24 independent

inadvertently generated events with these characteristics is shown in Figure 21. This demonstrates that similar inputs do not necessarily lead to similar outputs where there is modeling uncertainty. Caution is needed to not draw an overly optimistic conclusion from results such as the probabilistic density function plot Fig. 21a. Abnormal-environment representation of individual events is an uncertain process, and so is the conjunction. Dependence effects (addressed in Section 3 of this report) add another important factor to the uncertainty.

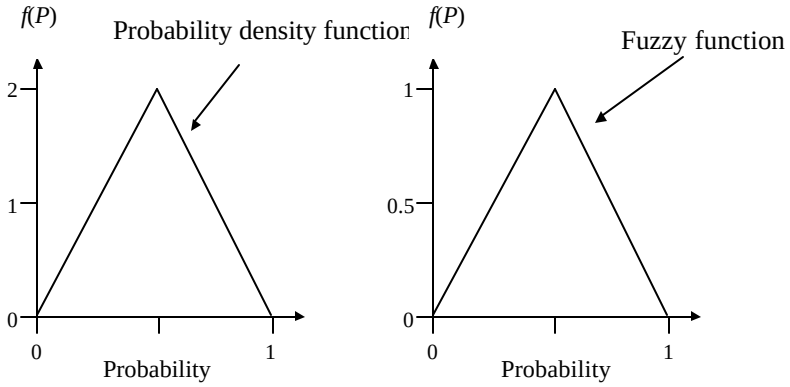

Figure 20. Two Representations of Inadvertently Generated Event Uncertainty

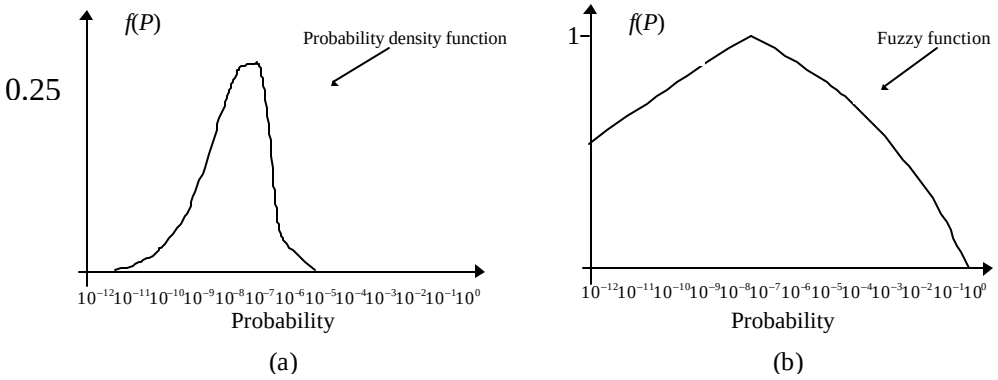

Figure 21. Corresponding Representations of 24-Independent-Event Conjunction

Appendix E: Modern Linear Algebra Concepts

The basic mathematical techniques used in pseudorandom generators and the Sandia National Laboratories UQS one-way transforms are from a branch of mathematics that is widely used in applications such as error-protection coding and cryptology, but are not familiar to practitioners in many other branches of science. For this reason, some applicable background will be outlined in this appendix. The descriptions below are terse and some are mathematically imprecise and incomplete. For a more detailed explanation, see Ref. 8.

A *Group* is a structure with one operator that satisfies closure and associativity. It includes an identity element and an inverse for each element. The exclusive-or operation (along with all other modulo operations) is a group.

A *Ring* has two operators. The ring is a commutative group under the "add" (not necessarily arithmetic add) operator. It has closure and associativity under "multiplication" (not necessarily arithmetic multiplication). "Multiplication" distributes over "addition." Multiplicative inverses are not assured.

A *Field* is a ring with commutative "multiplication" having a multiplicative identity and an inverse for every element except the additive identity. A field that is restricted to elements 0 and 1 and has operators exclusive-or (for "addition") and "and" (or an equivalent) for "multiplication" is called GF(2) (Galois field of two elements).

A *Subgroup* is a subset of group elements with closure.

A *Vector Space* over a field is a commutative group under "addition." Vector space elements can be multiplied by field elements (including one-way field element distribution over vector addition and vector element distribution over field addition), there is associativity in one-way multiplying multiple field elements and a vector.

A *Linear Associative Algebra* extends a vector space by incorporating associative vector multiplication along with bilinearity (vector multiplication distributes over vector addition, where the vector addition elements can be scaled through multiplication by field elements).

Vectors can be denoted by "*n*-tuples" (ordered collection of field elements). Addition of GF(2) *n*-tuples is done by adding field elements from corresponding positions using exclusive-or. "Inner-product" multiplication of *n*-tuples in matrix operations uses exclusive-or addition of elements obtained by multiplying elements from corresponding positions. Two vectors with inner product zero are said to be "orthogonal."

A *Vector Subspace* is a subset of vector elements having closure.

An $n \times m$ *Matrix* is an ordered set of nm elements in an array of n rows and m columns. The one-way transform matrices used by Sandia National Laboratories contain GF(2)

elements. A nonsingular matrix is an $n \times n$ matrix that has an inverse such that their product is a matrix containing all 1s where the row number and column number are equal, and all zeros elsewhere.

An *Ideal* is a subset of ring elements and is an additive subgroup, such that products of ideal elements with ring elements yields ideal elements.

A *Polynomial* can have elements from any field. The highest order of the polynomial variable is called the "degree" of the polynomial. In most pseudorandom generators, polynomials are from GF(2) and the degree is denoted by "*n*." Addition and multiplication of polynomials is a ring. A polynomial of degree *n* not divisible by any polynomial of degree less than *n* is irreducible. A set of polynomials is an ideal if and only if it consists of all multiples of a generator polynomial. The ring of polynomials is a principle ideal ring. The residue class ring is all multiples modulo the generator polynomial (remainders on division by the generator). The residue classes of polynomials modulo a generator polynomial of degree *n* form a linear associative algebra of dimension *n*.

A *Galois Extension Field of 2^n Elements* is an algebra of polynomials modulo an irreducible polynomial of degree *n*, denoted GF(2^n) (Galois field of 2^n elements).

A *Multiplicative Group of a Galois Extension Field* is an element and all of its powers modulo a polynomial (a cyclic group of some number of elements is called an "order"). If the polynomial is "primitive," the order is $2^n - 1$ (where an irreducible polynomial of degree *n* gives a group of $2^n - 1$ elements).

References

1. Spray, S. D., and J. A. Cooper, “The Unique Signal Concept for Detonation Safety in Nuclear Weapons,” Sandia National Laboratories Report SAND91-1269, June 1993.
2. Cooper, J. A., “Separate-Event Unique Signal Transmission,” Sandia National Laboratories Report SAND90-0315, December 1991.
3. Cooper, J. A., “Dependence Effects in Unique Signal Transmission,” Sandia National Laboratories Report SAND88-0394, April 1988.
4. Ash, Carol, *The Probability Tutoring Book*, IEEE Press 1993 pp. 266-268.
5. Hamming, Richard, *The Art of Probability,* Addison-Wesley Publishing Co. 1991.
6. Leyland, Jacqueline, “Unique Signal Analysis and Synthesis Project,” Master’s Thesis, Duke University, April 24, 1996.
7. Shannon, C. E., *“A Mathematical Theory of Communication*," Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, July and October, 1948.
8. Peterson, W. W., and E. J. Weldon, Jr., *Error-Correcting Codes*, The MIT Press, 1972.
9. Von Mises, Richard, *Mathematical Theory of Probability and Statistics*, Academic Press 1964.
10. Ahmed, N. and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing*, Springer-Verlag 1975.
11. Cooper, J. A., “An Assessment of Mixing Multiple-Source Unique Signals,” Sandia National Laboratories Report SAND89-2910, June 1990.
12. Diffie, W., and M. E. Hellman, “New Directions in Cryptography,” *IEEE Transactions on Information Theory*, IT 22, No. 6 1976.
13. Tremblay, J. P., and R. P. Manohar, *Discrete Mathematical Structures with Applications to Computer Science*, McGraw-Hill 1975.
14. Cooper, J. A., “Digital Pattern Monitoring,” Sandia National Laboratories Report SAND77-0835, May 1977.
15. Cooper, J. A., “The Application of Chained One-Way Transforms to Intent Unique Signal Transmission and Verification,” Sandia National Laboratories Report SAND87-2741, February 1988.
16. Cooper, J. A., “One-Way Transformation of Information,” United States Patent No. 4,841,570, June 1989.
17. Haasl, David, Norman Roberts, William Vesely, and Francine Goldberb, *Fault Tree Handbook*, U.S. Nuclear Regulatory Commission 1981.
18. Kaplan, Wilfred, *Advanced Calculus*, Addisson-Wesley 1956.
19. Sveshnikov, S. S., *Problems in Probability Theory, Mathematical Statistics, and Theory of Random Functions*, Dover Publications 1978.
20. Kaufmann, Arnold, and M. M. Gupta, *Introduction to Fuzzy Arithmetic*, Van Nostrand Reinhold 1991.
21. Cooper, J. A., Scott Ferson and Lev Ginzburg, "Hybrid Processing of Stochastic and Subjective Data," *Risk Analysis Journal*, December 1996.
22. Ferson, S. *RAMAS Risk Calc 4.0 Software: Risk Assessment with Uncertain Numbers*. Lewis Publishers, Boca Raton, Florida 2002.

Author Biography

Arlin Cooper (Ph.D., Electrical Engineering, Stanford University, 1964; M.S.E.E. and B.S.E.E., University of New Mexico, 1958 and 1957, respectively) is a Distinguished Member of Technical Staff at Sandia National Laboratories, where he has been for 39 years. His experience includes 10 years working on nuclear weapons safety components, five years working on unique-signal-processing methodologies, 10 years working on mathematical assessment methodologies in the nuclear weapons safety assessment area, and four years supporting safety assessment as a matrixed activity out of the Airworthiness Assurance Department.

During the first 10 years of this time, he was supervisor of various organizations, including the organization that produced the Sandia National Laboratories EMP handbook, which he edited and for which he wrote parts; the organization that showed the feasibility of a surface-arc lightning arrester connector, for which he received a patent; and the organization that produced the first TSSGs, for which he designed several of the algorithms used, including the first one-way transform. During the five years working on unique-signal-processing methodologies, he designed the "level-two" and "level-three" one-way transforms, receiving a patent for the latter, and helped represent Sandia National Laboratories unique signal safety interests to the AMAC POG. During the 10 years in safety assessment (including eight years working for Stan Spray), he identified previously undiscovered safety shortfalls in about a dozen unique-signal-processing algorithms, and derived remedies. He also developed the PHASER (Probabilistic Hybrid Analytical System Evaluation Routine) and COSMET (Coordinated Objective/Subjective Mathematically Enhanced Tools) uncertainty analysis methodologies, for which approximately 180 users in 17 countries have requested the software, and taught approximately 100 short courses to Sandia National Laboratories personnel on safety analysis methodology in the NST (National Surety Training) program and in the weapons intern program. During the past four years, he has designed a number of unique signal patterns and unique-signal-processing architectures for various weapon safety improvement programs. He has written three books, contributed to several others, and has written about 200 journal articles, conference papers, and formal reports.

Glossary

Abnormal environments: Environments in which the system is not required to retain full operational capability, although it must retain full safety capability[51]. Some also interpret these as environments defined as abnormal in the weapon STS, which is a less safety-conservative definition.

AMAC POG: Aircraft Monitor and Control Project Officer's Group, an instrument of the Chief of Staff of the Air Force, the Secretary of the Air Force, and the DoD to standardize, coordinate, publish, and maintain interface and test criteria for assuring compatibility between National Nuclear Surety Administration (NNSA) developed weapons and aircraft/sir launched delivery systems.

Dependence: Interrelationship between entities, such as having a common constituent (common-mode effect), or having some feature of one entity that is affected by some feature of another entity.

DoD: Department of Defense, the "customer" for weapon systems.

ENDS: Enhanced Nuclear Detonation Safety, the approach instituted in the early 1970s to produce a coordinated safety theme based on the principles of isolation, inoperability, and incompatibility, which included the unique signal.

Event: A communicated action that can be interpreted as one independent entity of a unique signal.

Fuzzy uncertainty: Subjective uncertainty defined mathematically by a function that captures ranges of values in an abscissa as a function of amount of presumption in expert judgment represented as an ordinate, such that ranges represented by higher presumption are nested within ranges represented by lower presumption, and so that the maximum amount of presumption is normalized to one.

Intent: An unambiguous human-initiated sequence of events (a UQS) that follows authorization to use a nuclear weapon and signifies that the weapon can be pre-armed for use.

Hexadecimal: A base-16 number system, sometimes used on data-entry keyboards and in other places where numeric data are processed.

LFSR: Linear feedback shift register, a recursive shift register where feedback is determined by linear (GF) logic elements, *e.g.*, exclusive-or logic.

[51] It is important to note that the system may retain much of its operational capability, which can be more dangerous than if all operational capability were lost.

MCs: A documentation of military characteristics that defines the required properties and functionality of a weapon system.

Model: A mathematical function that is intended to approximate an actual function as closely as possible.

Normal environments: Environments under which the system is required to retain full operational capability and full safety capability. These are sometimes viewed as those environments represented as normal in the weapon STS, but this is a less safety-conservative viewpoint.

Pattern: The specific sequence of unique signal event types that signifies human intent to pre-arm a safety subsystem.

PDF: Probability density function that uses the ordinate to represent likelihood of abscissa values.

Possibilistic uncertainty: Subjective uncertainty defined mathematically by a function that captures ranges of values in an abscissa as a function of amount of presumption in expert judgment represented as an ordinate.

ROM key: A read-only memory device for unique signal pattern entry, such that the key cannot be inadvertently inserted (*e.g.*, safety-wired in an inclusion region), and such that the events are read sequentially from the memory.

Separate-event communication: Communication of unique signal events in a manner that disassociates each event from the others insofar as practical (for example, by transmitting each event in a separate message of a communication protocol).

Stronglink Switch: An abnormal-environment-resistant responder that enables weapon arming and/or firing signal communication in response to a correct unique signal, but which locks up on any incorrect event.

STS: Stockpile-to-target sequence for situations leading to environments in which a weapon may be subjected during its lifetime.

System 2: A specification for digital communication between aircraft and "stores."

TSSG: Trajectory Sensing Signal Generator.

Type: A distinguishing property used to discriminate two or more (preferably two) possible characteristics for each communicated event.

Uncertainty: Incomplete knowledge about a mathematical parameter or a mathematical model.

UQS: Shorthand representation for “unique signal.”

Variability: Objectively known variation due to statistically represented changes (*e.g.*, the number thrown on a fair die).

Weaklink: A component necessary for weapon operation that irreversibly fails in correspondence to first-principles of physics or chemistry in environments less severe than those that could bypass stronglink or exclusion-region components.

WH Interface: The interface between the warhead and weapon components that are the responsibility of other organizations (*e.g.*, the DoD).

Index

DISTRIBUTION:

Carlson, David D.	12300	MS 0428
Johnson, Victor J.	12301	MS 0428
McCulloch, William H.	12300	MS 0428
Spray, Stanley D.	12300	MS 0428
Stevens, William L.	12300	MS 0428
Trauth Jr., Charles A.	12301	MS 0428
Sjulin, Janet M.	12323	MS 0829
Spencer, Floyd W.	12323	MS 0829
Blackledge, Michael A.	12326	MS 0638
Olson, David R.	12332	MS 0492
Chen, Kenneth C.	12332	MS 0492
Cincotta, Harry L.	12332	MS 0492
Dvorack, Michael A.	12332	MS 0492
Mahn, Jeffrey A.	12332	MS 0492
Maloney, Kevin J.	12332	MS 0492
Mauldin, Ed	12332	MS 0492
McCollister, Daryl R.	12332	MS 0492
Summers, Daniel A.	12332	MS 0492
Wolcott, James F.	12332	MS 0492
Jones, Todd R.	12333	MS 0405
Baca, Rob. G.	12333	MS 0405
Bohn, Michael P.	12333	MS 0405
Brown, Thomas D.	12333	MS 0405
Camp, Susan E.	12333	MS 0405
Fuentes, Martin K.	12333	MS 0405
Lin, Yau Tang	12333	MS 0405
Pedersen, Ronald D.	12333	MS 0405
Schriner, Heather K.	12333	MS 0405
Sobolik, Keri B.	12333	MS 0405
Diegert, Kathleen V.	12335	MS 0830
Hoffman, John P. Jr.	12345	MS 0491
Stichman, John H.	2000	MS 0457
Novotny, George C.	2001	MS 0457
Rottler, Stephen J.	2100	MS 0429
Hartwig, Ronald C.	2100	MS 0427
Lucy, Tana B.	2102	MS 0435
Sanders, Gary A.	2103	MS 0453
Harrison, James O.	2111	MS 0447
Hoover, Phil D.	2111	MS 0447
Hillhouse, Aaron L.	2112	MS 0483
Tedeschi, William	2113	MS 0479
Caldwell, Michele	2113	MS 0479
Rosenthal, Mark A.	2114	MS 0481
Thomas, Danny L.	2114	MS 0481
Meeks, Kent D.	2131	MS 0482

Ortiz, Keith	2131	MS 0482	
Callahan, Michael W.	2300	MS 0509	
Plummer, David W.	2330	MS 0503	
Molley, Perry A.	2331	MS 0537	
Brandt, Dale J.	2331	MS 0537	
Laguna, George R.	2333	MS 0533	
Weiss, Douglas R.	2333	MS 0533	
Eilers, Dennis L.	2339	MS 0503	
Cooper, Harold L.	2339	MS 0503	
Deming, Douglas M.	2339	MS 0503	
Kreutzfeld, Richard E.	2613	MS 0319	
Eras, Kenneth.	2613	MS 0319	
Greenwood, William H.	2613	MS 0319	
Randall, Gary T.	2613	MS 0319	
Robinson, Jeffrey A.	2613	MS 0319	
Vanecek, Charles W.	2613	MS 0319	
Peter, Frank J.	2614	MS 0329	
Shirley, Clinton G.	2820	MS 0453	
Murphy, Melissa J.	2900	MS 0469	
Shaw, John D.	2911	MS 0631	
Rogulich, Andrew J.	2911	MS 0632	
D'Antonio, Perry E.	9713	MS 0136	
Tatro, Marjorie L.	6200	MS 0741	
Perry, Richard L.	6252	MS 0615	
Cooper, J. Arlin	6252	MS 0490	(35)
Covan, John M.	6252	MS 0490	
Ekman, Mark E.	6252	MS 0490	
Kuswa, Glen W.	2954	MS 0635	
Dalton, Larry J.	2662	MS 0860	
McCaughey, Kathleen G.	14400	MS 0868	
Cranwell, Robert M.	15312	MS 1176	
Robinson, David G.	6413	MS 0748	
Camp, Allen L.	6410	MS 0747	
Wyss, Gregory D.	6410	MS 0747	
Trucano, Timothy G.	9211	MS 0819	
Baty, Roy S.	2131	MS 0482	
Ringland, James T.	8112	MS 9201	
Henson, Douglas R.	8200	MS 9007	
Miller, Russell G.	8221	MS 9007	
Van Cleave, Randall A.	8221	MS 9034	
Talbot, Edward B.	8222	MS 9036	
Hinckley, C. Martin	8231	MS 9007	
Wichman, Elizabeth C.	8222	MS 9036	
Gehmlich, Douglas L.	8241	MS 9014	
Johnson, Alice J.	8241	MS 9014	
Monson, Robert D.	8243	MS 9108	

Molle, Raphael M.	8241	MS 9014
Cashen, Jerry J.	8205	MS 9202
Schroeder, Donald H.	9411	MS 0630
Central Technical Files	8945-1	MS 9018
Technical Library	9616	MS 0899 (2)
Review & Approval Desk (For DOE/OSTI)	9612	MS 0612