SIMULATION-BASED TECHNIQUES FOR CALCULATING FAULT RESOLUTION AND FALSE REMOVAL STATISTICS

Eric Gould
DSI International
1574 N. Batavia #3
Orange, CA 92867
(714) 637-9325
egould@dsiintl.com

Danver Hartop
DSI International
1574 N. Batavia #3
Orange, CA 92867
(714) 637-9325
dhartop@dsiintl.com

Abstract - This paper will discuss the use of diagnostic simulations to generate the Fault Resolution metric for a system or equipment. Simulation-based calculations are free of some of the biases that inhere within traditional, math-based approaches. Moreover, a simulation-based evaluation of the replacement of failed items also provides a basis for the calculation of the effect of diagnostic ambiguity upon false removals—including the estimated costs that can be attributed to removals beyond those that would be expected during a product's intended lifetime.

Introduction

In MIL-STD-2165 (Testability Program for Electronic Systems and Equipments), two separate equations are provided for the calculation of Fault Resolution—a quantitative Testability metric that measures how well a diagnostic sequence or strategy is able to isolate to fault groups that can be repaired with no more than a given number of replacements [1]. Each of these two equations corresponds to a different maintenance philosophy. The first equation—which is used when the selected system or equipment is maintained using block replacement (where all items in each isolated fault group are replaced or repaired at the same time)—has, with a few minor changes, become the most widely used method for calculating this metric within Testability analyses. The second equation—which attempts to quantify the Fault Resolution that results when a system is maintained using serial or prioritized replacement (where the components in isolated fault groups are replaced one at a time with intervening retesting)—has also been used to generate this metric. MIL-HDBK-472 (Maintainability Prediction) also provides two equations to be used when assessing a test sequence or strategy's ability to unambiguously isolate malfunctions: Percent isolation to a single replacement item (RI) and Percent isolation to a group of RIs [2]. Although these equations do not explicitly address different maintenance approaches, they can be interpreted in a manner that's applicable for systems maintained using either block or serial replacement.

These two sets of equations are representative of the various math-based techniques that have been used to calculate a single metric—usually referred to as either Fault Isolation or Fault Resolution—that quantifies the extent to which the diagnostics for a given system or equipment is able to unambiguously isolate malfunctioning items and thereby eliminate superfluous maintenance actions. Often expressed as a "weighted percentage" (a percentage of the overall system failure rate), this metric is typically calculated by summing failure rates for all fault groups for which a failure can be corrected with the specified number of replacements.

Although the equations in both MIL-STD-2165 and MIL-HDBK-472 call for the summing of component (module or part) failure rates, it has long been known that these calculations are more accurate when the failure rates are broken down by function or failure mode. Moreover, for diagnostic strategies that can isolate a single function or failure mode to more than one different fault group (such as multiple-failure strategies, which are designed to accommodate the "masking" effect that can often result from multiple, simultaneous malfunctions), the accumulated failure rates are often broken down

even more. Regardless, however, of the level at which the accumulating is performed, this metric is nearly always calculated by examining all fault groups that can be isolated by the diagnostics and summing the appropriate failure rates for the fault groups that meet the given criteria. Within this paper, we shall refer to calculations of this type as *traditional* or *deterministic* approaches to calculating the metric.

Three Problems with Traditional Fault Resolution Calculations

Now, there are several inherent problems with the traditional methods of calculating Fault Resolution that can result in serious inaccuracies when the resulting values are used to predict actual diagnostic performance. First of all, traditional calculations are *ahistoric*—statistics are calculated across the entire set of possible fault groups without any concern for the order in which the components might fail. At first glance, this would seem to be a good thing—after all, we wouldn't want our diagnostic predictions to be tied to the assumption that individual components will invariably fail in a predetermined order. Ahistoric calculations of this type, however, are based upon an invalid assumption—that the functions isolated within each fault group will, over time, tend to fail in accordance with their respective failure probabilities. On the contrary, most complex systems contain many functions that do not fail anywhere near as frequently as their reliability estimates indicate, regardless of how long the system is fielded. The reason for this lies not in the reliability of these functions, but rather in the relative unreliability of other functions—both other functions of the same components and other functions that reside in the same replacement group.

Imagine, if you will, a part that is comprised of two functions (function A and B), one of which (A) fails twenty times more frequently than the other (B). Because the entire component is replaced each time that function A malfunctions, function B is not given the chance to fail. Although a failure to function B is still possible, its relative likelihood would be greatly diminished (in many systems, this function would simply never fail). Generally speaking, the failure frequency experienced for the different functions on a multi-function component will not be as high as the estimated failure rates for those functions would imply. Although the net effect upon Fault Resolution is more significant when there is a large difference in the failure rates of the individual functions, there can still be a substantial reduction in failure frequency when all functions of a component have the same failure rate (since some functional failures will be postponed each time that the item is replaced).

This same situation can arise when functions from different components exist in the same replacement group. Take, for example, two functions (X and Y) that have significantly different failure rates (say, X fails twenty times more frequently than Y), and yet, because they both reside in a difficult-to-access location, they are always replaced as a pair (a recommended practice if function Y were relatively inexpensive to replace). Failures to function Y would be perpetually deferred, since it would be replaced with each failure to function X.

Because deterministic Fault Resolution calculations do not take these situations into consideration, they are inevitably biased towards the functions that fail more frequently. Put another way, they fail to show that many reliable functions fail even less frequently than their assigned failure rates would imply. If these more reliable functions can be uniquely isolated, then the Fault Resolution statistics would be falsely skewed toward the smaller group sizes. If, on the other hand, these more reliable functions are isolated in larger groups, then the Fault Resolution statistics would predict a diagnostic performance that is substantially worse than that which would be actually realized in the field.

A second problem with traditional Fault Resolution calculations is that they can only forecast diagnostic performance over an arbitrarily large time interval. Failure rates, it must be remembered, are not predictions, but rather estimated means. Because the Fault Resolution is traditionally calculated by summing these means, it will not properly reflect the diagnostic behavior of a system until it has been fielded sufficiently long that the rates of actual failure occurrences have begun to approach their means. Before this will be true, all functions will have to have failed several times. Since many of the systems functions will inevitably be highly reliable, this means that the system may have to be fielded an arbitrarily large amount of time before it will approach the diagnostic performance predicted by traditional Fault Resolution calculations.

Because Fault Resolution is traditionally calculated using all failures in all possible fault groups, they will inevitably include functional failures that are highly unlikely during the expected lifetime of the system. This results in an additional bias toward functions that fail frequently. Once again, if these functions are isolated within relatively small fault groups, then the

Fault Resolution figures will be more attractive than the diagnostic performance that is actually achieved during the system's expected lifetime. On the other hand, if these highly reliable functions are isolated within relatively large groups, then the prediction will be less attractive than actual system performance.

A third problem with traditional, deterministic Fault Resolution calculations is that they do not distinguish between the different failure combinations that will result in isolation to the same fault group. This is not a problem with either single-fault isolation strategies (that is, strategies that assume that only a single function can be malfunctioning as the system is diagnosed) or systems that are maintained solely using block replacement (or better yet, systems for which all failures can be unambiguously isolated to a single malfunctioning component). When, however, a system is diagnosed using a multiple-fault isolation strategy (one that is able to accommodate multiple, simultaneous malfunctions), then serial replacement of suspected items may result in different diagnostic behavior when the same fault group is isolated due to different combinations of failed functions. If the replacement of an item would result in isolation to a different fault group—even if it's a subset of the originally isolated group—then a fault has been successfully resolved. A fault group might, for one failure combination, require two replacement actions to observably correct a failure; the same fault group, however, for a different failure combination, might observably correct one failure with one replacement and another failure with a second replacement. Since there is no ambiguity involved for this second failure combination (each replacement observably corrects a malfunction) this would be recorded as two replacements of one item each (rather than as one replacement of two items).

Traditional Fault Resolution calculations, once again, do not distinguish between the different diagnostic behavior that would result when a fault group can be isolated due to different failure combinations. When a multiple-failure strategy is used to diagnose a system that is maintained using serial replacement, traditional ways of calculating Fault Resolution will result in a bias toward the larger group sizes.

Several Advantages of Monte Carlo Diagnostic Simulations

Diagnostic simulations have long been used to calculate Reliability and Maintainability predictions. Their application within Testability analysis, although similar in conception, has been less common due in part to entrenched engineering practices. Testability analyses, although most profitably employed in early phases of the design process, have frequently been dismissed as mere measures of contract compliance and subsequently postponed until the design has been more or less fixed and most reliability concerns have already been addressed. When Testability has indeed been examined early in the design process, analysts have at times assumed that, because the figures are calculated using preliminary data, any effort spent to achieve additional accuracy would result in diminishing returns. Assumptions like these—exacerbated, perhaps, by the difficulty of finding a diagnostic engineering tool that generates simulation-based metrics—have prevented the Testability community from taking advantage of the same techniques that have met with such success in the Reliability and Maintainability worlds.

Monte Carlo simulations—in which individual events are generated randomly, yet in accordance with their assigned rates and distributions—are ideally suited for the simulation of diagnostic behavior. First of all, they can be easily adapted to accommodate any desired isolation or maintenance philosophy. For analyses that assume single-point failures, the diagnostic strategy would be immediately invoked each time a malfunction occurs. For multiple-failure strategies, on the other hand, the simulation can account for any delay that may exist between a failure and its diagnosis. During this delay—which can either be a fixed interval (as would be the case with scheduled maintenance) or probabilistic (when the interval between runs of the diagnostics falls into a distribution of its own)—other malfunctions might occur. Of course, the longer the delay between diagnostic sessions, the greater the likelihood that multiple, simultaneous malfunctions will exist as the diagnostics are performed. In order to simulate the real-life deployment of the system or equipment, mission profiles can be developed that account for different states in which the system might exist, as well as the use of multiple diagnostic approaches (failure-driven, periodic, prognostic, etc.). Complex maintenance philosophies (combinations of block, serial and hybrid replacement strategies) can also be easily handled by a Monte Carlo simulation.

The random number generator that is at the heart of all Monte Carlo simulations can easily accommodate any number of standard failure distribution curves. Testability metrics (such as Fault Resolution) have traditionally been calculated by adding up estimated item Failure Rates. Each Failure Rate—whether

derived from empirical data or from theoretical projection—is the multiplicative inverse of the item MTBF (Mean Time Between Failures) that has been scaled to represented the expected mean number of failures per million hours. Regardless of the form in which it is expressed, this standard reliability metric is a mean that has been removed from its required statistical context. It is no longer possible to tell 1) the time interval over which the mean is expected to hold true, or 2) the specific distribution curve for which the mean was calculated. Because of this, MTBFs are much maligned in some quarters. Since Monte Carlo diagnostic simulations take into account not only the MTBF, but also the specific failure distribution curve for each component, the analyst is able to find answers to whole new types of questions, such as "What would be the net effect on our ability to isolate faults if we were to replace this component with one that is ***less*** reliable, yet will exhibit fewer infant failures?" Diagnostic simulations ultimately empower the analyst, allowing him or her to make sophisticated decisions based on realistic projections of the effects that small or subtle changes might have upon system performance.

Finally, because Monte Carlo simulations can be easily programmed using standard event processing algorithms, they can provide an elegant method for analyzing a wealth of secondary statistics by simply scanning through the events generated during a simulated run. In the Maintainability community, this capability has already been tapped to generate Life Cycle Cost and Operational Availability estimates. A sophisticated diagnostic simulation, however, might allow simulated events to be compared with events that, for one reason or another, were prevented from occurring during the simulation. For example, the set of simulated failures could be compared against the set of failures that would have occurred if diagnostic ambiguity had not resulted in false removals during the simulation. Using this capability, new simulation-based Testability metrics could be derived that show the *effects of diagnostic ambiguity* upon the Life Cycle Cost or Operational Availability of a system or equipment. Rather than act as predictions, these figures could be used to assess the *impact* of both design and testing decisions upon the diagnostic performance of the system or equipment. Unlike the corresponding Maintainability predictions, these new metrics could be generated in relatively early phases of product development.

Simulation-Based Fault Resolution and False Removal Calculations

When the Fault Resolution metric is calculated using data collected from a set of Monte Carlo diagnostic simulations, it is not subject to the biases that inhere within traditional Fault Resolution calculations. First of all, any properly-designed diagnostic simulation will automatically take into account the replacement history of each item when computing functional failures. When a component is replaced—regardless of whether the part had actually failed, was falsely removed due to diagnostic ambiguity, or was intentionally replaced prematurely as the result of a prognostic decision—the next failure to each function of that component will be recalculated from that point forward. In other words, each simulated functional failure takes into consideration the maintenance history (in particular, the elapsed time since the most recent replacement) of its respective component.

Because diagnostic simulations can be performed over any desired time interval, the resulting statistics do not assume an arbitrarily long deployment. On the contrary, a short simulation might be run to ascertain the types of failures that might be expected during, say, the first year that a system is deployed. Because simulated malfunctions occur not merely in accordance with the component MTBF, but also with its assigned distribution curve, a certain degree of randomness is preserved. To minimize the negative effect of this randomness, the final metric represents an average of data that has been gathered from a large number of individually simulated lifetimes. To increase the accuracy of the calculation, additional simulation runs should be averaged into the metric. Increasing the time of the simulation, on the other hand, will result not in more accurate numbers, but rather in estimated performance over a longer time interval. If a given system or equipment is expected to be fielded for thirty years, then metrics that are generated using simulated lifetimes longer than that would erroneously account for failures that would not be likely to occur during the useful life of the system. Unlike traditional Fault Resolution calculations (which, remember, are sums of means) the accuracy of simulation-based calculations is not constrained by the time intervals over which component MTBFs are assumed to hold true. Furthermore, because simulations allow the same metric to be calculated over several different time intervals, analysts can formulate long-term maintenance plans, rather than assume that one

value encapsulates all knowledge about the future diagnostic behavior of the system.

Finally, for each isolated fault group, the simulation knows the particular combinations of failed functions that resulted in the isolation of that group. This allows the simulation to take anticipated diagnostic behavior into consideration when estimating the number of replacements needed to correct a failure. If, for a serially-replaced fault group, retesting results in isolation to a different group (a subset of the original group) after a single component has been replaced, then a fault has been corrected with one replacement—even though additional components within the originally isolated fault group will have to be replaced before the system is fully repaired.

Moreover, because the diagnostic simulation knows not only the specific failure combination that drives each fault isolation, but also the specific replacement algorithm (block, serial, hybrid) that will be used to repair the isolated fault group, false removal rates (or, to be more precise, the *estimated effect of diagnostic ambiguity upon false removals*) can be calculated using data from the diagnostic simulation. These false removal rates cannot be accurately estimated using traditional, deterministic methods. The simulation tool, by comparing the number of simulated false removals for each component with the number of times that the part would fail if it were not prematurely replaced, can determine the expected number of extra replacements that will be required over the useful life of the system. Extra replacements refer to replacements beyond those that would be necessary if there were no diagnostic ambiguity and no preemptive prognostics. Taking this one step further, a software engine could then easily calculate the average cost resulting from extra replacements—a measure of *the effect of premature replacement upon the Life Cycle Cost* of the system.

Example Fault Resolution Calculations Using the *eXpress* Simulation Engine

In this section, we will present some examples of simulation-based Fault Resolution calculations that have been generated using the diagnostic module of ***eXpress***—a diagnostic engineering tool developed by DSI International.

Within ***eXpress*****,** a distinction is made between Fault Isolation and Fault Resolution. Fault Isolation refers to the process of overlaying test information in order to reduce the set of suspected components (those within which a fault has been detected) to the smallest group guaranteed to contain a failure. Fault Isolation statistics, then, quantify the ability of the diagnostic strategy to unambiguously isolate faults for a given system or equipment. This metric—which ***eXpress*** calculates using traditional, equation-based techniques—closely corresponds to traditional Fault Isolation and Fault Resolution calculations when it is assumed that isolated fault groups will be maintained using block replacement.

Fault Resolution (at least as it is handled within the ***eXpress*** tool), quantifies a diagnostic strategy's ability to resolve failures for a given design. The Fault Resolution metric—which ***eXpress*** generates using a Monte Carlo diagnostic simulation—takes into consideration both the particular diagnostic strategy and the maintenance philosophy (block, serial, hybrid) that has been defined for each isolated fault group. Fault Resolution statistics measure the ability of the specified diagnostics and maintenance plan to unambiguously resolve failures for a given system or equipment.

The following examples will compare Fault Isolation and Fault Resolution statistics as they are calculated by the ***eXpress*** diagnostics module. In order to highlight the biases that are corrected when diagnostic simulations are used in place of traditional techniques, these examples will be set up to exploit the particular feature being demonstrated. For real-life systems, the effect upon the final statistics may or may not be greater than that demonstrated in these examples. Differences between Fault Isolation and Fault Resolution are dependent upon a large number of factors (the topology of the design, the distribution of failure rates with respect to fan-outs within the design, and the relative frequency of failures to different functions on the same item—just to name a few). For larger systems, of course, the statistical differences between Fault Isolation and Fault Resolution are not likely to be attributable to any single factor or component; instead, they will be the cumulative effect of many factors associated with a large number of components.

Example 1: Block Replacement

In our first example, we will examine the effect that the premature replacement of an item (due to fault groups being replaced as a block) has upon other failures of that item. The following design (Figure 1) will be used for this example:

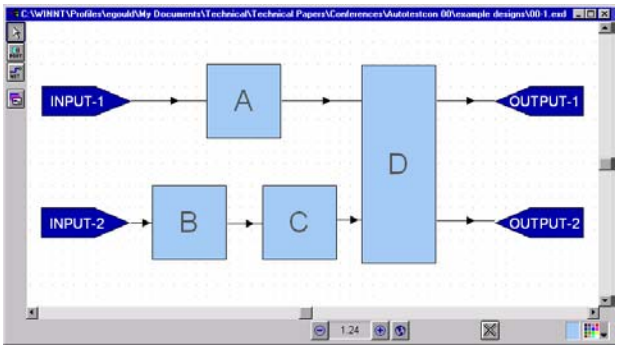


Figure 1. Design for Example 1

In this design, there are four replacement items (A, B, C & D) and five functions (one each for A, B & C; two for D). It is assumed that the only tests possible for this design are those performed at the two outputs. The test at OUTPUT-1 verifies the operation of item A and the first function of item D. The test at OUTPUT-2 verifies the items B & C and the second function of item D. For this example, each of the four items is considered equally likely to have failed—the Failure Rate is set to 1000 failures per million hours (about once every forty-one days)—and the two functions on item D are considered equally likely to fail.

When diagnostics are performed on this design, all failures can be isolated into one of two fault groups—one containing items A & D, and the other containing items B, C & D. The resulting Fault Isolation metrics for this design are summarized in the following table:

Table 1. Fault Isolation Metrics

Size of Isolated Fault Group	Probability	Cumulative Probability
2	37.50%	37.50%
3	62.50%	100.00%

According to these figures, 1½ out of every 4 failures (37.5%) are isolated to the group of size two and 2½ out of every 4 failures (62.5%) are isolated to the group of size 3. This is directly proportional to the number of items that are isolated within each group (if we count each function of D as ½). Since failures to each item are shown to be equally probable, this might at first appear to be an accurate indication of the relative frequency that each fault group would be isolated for this design.

Now, let's take a look at the Fault Resolution metrics (Table 2) that are produced within the ***eXpress*** diagnostic module, using data generated by a Monte Carlo diagnostic simulation. This simulation—which was run for 1000 simulated missions of 10,000 hours each—was set up so that diagnostics are performed immediately as each malfunction occurs (practically guaranteeing single-point failures) and that each isolated fault group will be replaced as a block.

Table 2. Fault Resolution Metrics

Replacements to Repair Isolated Fault Group	Probability	Cumulative Probability
2	46.24%	46.24%
3	53.76%	100.00%

For this design and these failure rates, malfunctions would be resolved by two replacements almost 9% more often than indicated by traditional Fault Isolation/Resolution statistics. To understand why this is so, we must first look at the number of failures that the simulation registered for each function (Table 3):

Table 3. Simulated Failures and Replacements

Functions	Average Failures	Average Replacements
A	8.42	9.85
B	5.04	11.45
C	4.96	11.45
D1	1.43	21.30
D2	1.45	

Notice that, although all four items have same failure rate, they did *not* fail equally during the simulation. This is not due to randomness or inaccuracies within the diagnostic simulation, but rather because items are prematurely replaced (due to block replacement) each time that a failure is repaired. If there were no

premature replacements, then each item would fail about 10 times within the simulated mission. With premature replacements, however, each item fails somewhat less than its failure rate would indicate (since each time that it is replaced before it fails, the next failure to that component is postponed). Item A, because the only time in which it is prematurely replaced is when the first function of D fails, fails the most. At the other extreme is item D. Because D is replaced each time that any of the other items fails (remember, D is called out in both fault groups), actual failures to one of D's functions are relatively rare—only 2.88 times (the sum of the average failures for D1 and D2) within the simulated interval. Items B and C, on the other hand, fail about the same number of times—which is considerable less than for item A, since fewer failures to other items result in A's premature replacement.

With this in mind, it is easy to understand why failures are isolated to the fault group of size two nearly 9% more frequently in the simulation than in traditional Fault Isolation calculations. Each time that a malfunction is isolated a fault group, the block replacement of the other item(s) in the group results in the postponement of failures for the items in that group. Since the fault group of size three results in more false removals, failures within that group will tend to be deferred more than failures to the group of size two.

Let's summarize what we have seen so far. Isolation to larger fault groups, even though they tend to have higher failure rates, results in the postponement of more failures than does isolation to fault groups containing fewer items. This means that traditional Fault Resolution calculations (what ***eXpress*** calls Fault Isolation), because they do not account for this postponement, are biased toward the larger fault groups. In addition to this bias is the fact that when a part is replaced, failures to other functions of that part are postponed. This means that traditional calculations are also biased towards items whose functions are isolated within multiple fault groups (since, when an item is replaced, these calculations don't account for the postponement of failures to functions of that item that would be isolated to the other groups). Depending upon the size of the groups containing the other functions of the replaced item, this bias could favor either larger or smaller fault groups. Simulation-based Fault Resolution metrics do not exhibit either of these biases.

Let's complicate this example a little by changing the failure rates. Now, instead of each item, let each *function* have the same failure rate (once again, 1000 failures per million hours). This means that item D, which is the only item with two functions, will now have twice the failure rate of the other items.

Although the same fault groups are isolated by the diagnostics, the likelihood of isolating to each group is now different. Table 4 shows the Fault Isolation statistics that would now be produced by eXpress:

Table 4. Fault Isolation Metrics

Size of Isolated Fault Group	Probability	Cumulative Probability
2	40.00%	40.00%
3	60.00%	100.00%

Because each function has an equal failure rate, the fault group of size two (which can be isolated due to failures to 2 of the 5 functions) has an isolation probability of 40%. The fault group of size 3 contains 3 of the 5 functions and will therefore be isolated (according to this calculation) 60% of the time.

If we generate the Fault Resolution metrics, using the same settings as before, we get the following:

Table 5. Fault Resolution Metrics

Replacements to Repair Isolated Fault Group	Probability	Cumulative Probability
2	47.35%	47.35%
3	52.65%	100.00%

Although the difference between the traditional and simulation-based metrics is now under 7.5% (less than the nearly 9% difference we saw before), what is interesting is that the Fault Resolution metrics indicate that two replacements will correct a problem over 1% more frequently than in the previous case. The reason why this is interesting, however, may not be immediately apparent—we must once again look at the simulated failures and replacements for each function (Table 6).

Table 6. Simulated Failures and Replacements

Functions	Average Failures	Average Replacements
A	5.05	13.21
B	3.16	14.69
C	3.20	14.69
D1	8.16	27.91
D2	8.33	

Notice that, although D has only twice the estimated failure rate (2000 instead of 1000 per million hours), it now actually fails over five times more frequently (16.49 failures vs. 2.88 failures) within the simulated interval. Rather than letting failures to other items result in its own failures being postponed, item D fails first (since it now has twice the failure rate) thereby causing failures in the other items to be postponed. Rather than rarely fail, failures to D are now rarely deferred. Interestingly, however, item D is now replaced 27.91 times—only 6.61 times more than before. Replacements of item D have simply shifted so that, rather than be falsely replaced when other items fail, D is now replaced more due to its own failures. Nevertheless, failures to other items do at times cause D to be prematurely replaced—that's why item D fails only 16.49 times, 3.51 fewer times than the 20 failures that would be expected in this interval (according to the Failure Rate).

Because Monte Carlo diagnostic simulations can take into account not only the failure rates, but also the specific failure distributions for each component, the (simulation-based) Fault Resolution metrics are sensitive to differences in these distributions, even though traditional Fault Isolation calculations do not take failure distributions into account.

Using the same design, let's change the assigned failure distribution for item B from *standard normal* (the default within ***eXpress***) to *exponential*. We'll keep the same item Failure Rates (which represent the means of the distributions). Figures 2 and 3 show the example distribution graphs that are displayed within the ***eXpress*** object browser. Although both of these distributions may have the same failure rate, items with exponential failure distributions (due to their propensity for infant death) will tend to generate more failures during the simulated time interval than would items whose failures fall into normal distributions.

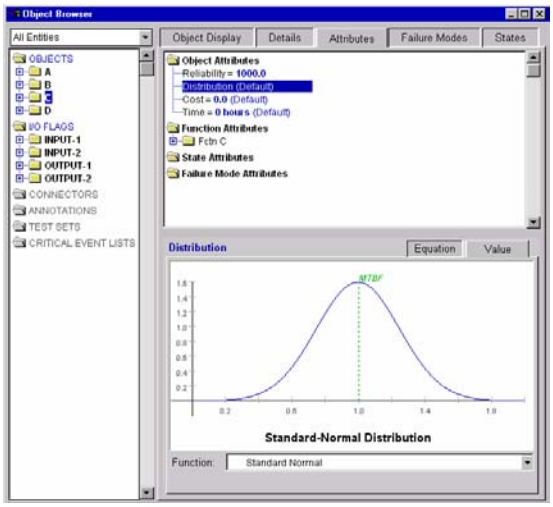


Figure 2. A Sample Standard-Normal Distribution in the ***eXpress*** Object Browser

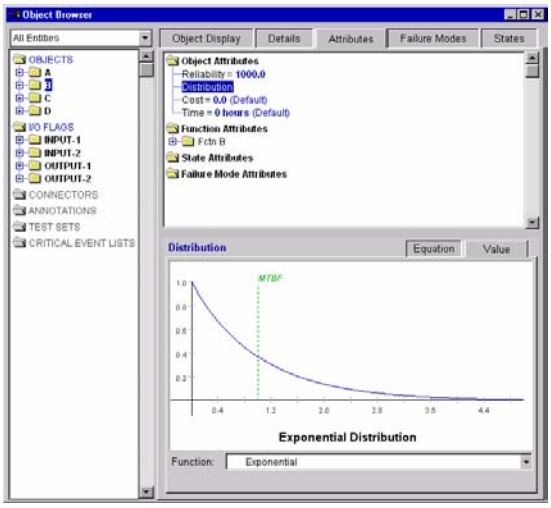


Figure 3. A Sample Exponential Distribution in the ***eXpress*** Object Browser

We need only take a quick look at the traditionally-calculated Fault Isolation statistics (Table 7) to see that they contain another bias, for the numbers are no different from those that were calculated when the standard normal failure distribution was used for item B:

Table 7. Fault Isolation Metrics

Size of Isolated Fault Group	Probability	Cumulative Probability
2	40.00%	40.00%
3	60.00%	100.00%

eXpress's Fault Resolution metrics (Table 8), on the other hand, tell a different story altogether:

Table 8. Fault Resolution Metrics

Replacements to Repair Isolated Fault Group	Probability	Cumulative Probability
2	38.80%	38.80%
3	61.20%	100.00%

Although these numbers could be compared with the Fault Isolation statistics for this set of distributions (Table 7), the fact that these two sets of numbers now more closely resemble each other should be a cause of some concern—the Fault Isolation statistics were not affected in any way by the change in failure distribution, whereas the simulation-based statistics reflected the projected difference in behavior.

More interesting is the comparison between the Fault Resolution metrics when item B was assigned the standard normal distribution (Table 5) and the metrics that result when item B's failures fall into an exponential distribution (Table 8). With item B's failures falling into an exponential distribution, the likelihood of repairing a malfunction with two replacements is nearly 10% less than when B's failures fell into a normal distribution (Table 5). That B is responsible for this change in statistics can easily be corroborated by looking the failures and replacements that occurred for each item in the diagnostic simulation (Table 9):

Table 9. Simulated Failures and Replacements

Functions	Average Failures	Average Replacements
A	5.68	12.73
B	10.02	20.08
C	2.91	20.08
D1	7.05	32.81
D2	7.15	

As we would expect, there has been significant increase in failures to item B relative to the other items that are replaced in its fault group (C & D). Interestingly, there has also been an increase in failures to item A relative to the other items. This is because A is the only item that is not prematurely replaced each time that a failure occurs for item B.

It should by now be apparent that the end effects of different failure rates & distributions can, for many systems, be quite unpredictable—it depends not only upon the topology of the design and the fault groups that can be isolated, but also on the failure rates and distributions assigned to other items in the design.

Example 2: Serial Replacement

In our second example, we will generate simulation-based Fault Resolution metrics for a system that is maintained using serial replacement (each time a fault group is isolated, one item in the fault group—in this example, the item most likely to have failed—is replaced and system is rediagnosed). The following design (Figure 4) will be used for this example:

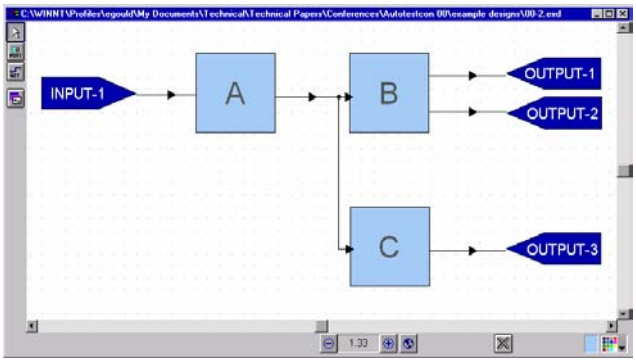


Figure 4. Design for Example 2

In this design, there are three replacement items and four functions (item B has two functions). Again, it is assumed that tests can only be performed at the outputs. Each of the three possible tests, then, verifies exactly two functions: A & B1 (OUTPUT-1), A & B2 (OUTPUT-2), or A & C (OUTPUT-3). The different failure distributions and MTBFs that have been assigned to each item are listed in Table 10 (for this example, the two functions on item B are assumed to be equally likely to fail):

Table 10. Assigned MTBFs for Example 2

Item	Failure Distribution	MTBF (in hours)
A	Std Normal	90
B	Std Normal	60
C	Exponential	135

For this design, ***eXpress*** isolates four functionally-unique fault groups (Table 11)—two of which,

though they contain different functions, are comprised of the same item:

Table 11. Isolated Fault Groups for Example 2

Isolated Functions	Isolated Items
A, B2	A, B
B2	B
B1	B
C	C

Because multiple-failure diagnostic strategies are designed to isolate any combination of simultaneous malfunctions, item A cannot be isolated in a fault group by itself (as it could if we were to assume single-point failures). As we shall see shortly, the use of serial replacement can compensate for any lost resolution due to the robust—or, if you wish, conservative—nature of multiple-failure isolation.

When calculated in the traditional manner, the Fault Isolation metrics for this design (Table 12) show that failures can be isolated to a fault group containing a single component a little over 60% of the time:

Table 12. Fault Isolation Metrics

Size of Isolated Fault Group	Probability	Cumulative Probability
1	60.53%	60.53%
2	39.47%	100.00%

As in the previous example, we will first compare these Fault Isolation metrics with Fault Resolution metrics derived using a diagnostic simulation. These statistics will be calculated using 1000 simulated missions of 10,000 hours each. For our first Fault Resolution calculation (Table 13), we will assume that the system is maintained using block replacement and that there is no diagnostic delay—in other words, malfunctions will be diagnosed as soon as they occur (this simulates the occurrence of single-point failures).

Table 13. Fault Resolution Metrics
(using Block Replacement)

Replacements to Repair Isolated Fault Group		Probability
1	66.12%	66.12%
2	33.88%	100.00%

In this example, the simulation-based statistics only offer about a 6% improvement over the traditional method of calculating the metrics. Because the diagnostics only isolate to a fault group containing a single component only 66% of the time, the Fault Resolution metric will reflect this relatively poor isolation when block replacement is used.

Let's rerun the simulation with one difference—we will allow the system to be maintained using serial replacement. For each isolated fault group, items will be replaced one at a time in a predetermined order. In this example, the replacement order will be based on the relative likelihood that a failure to each item results in the isolation of that fault group (when there is no diagnostic delay). After each replacement, the design is re-diagnosed to determine whether the replacement corrected a malfunction. The resulting Fault Resolution Metrics (Table 14) reflect the number of replacements needed to repair isolated fault groups using this maintenance philosophy.

Table 14. Fault Resolution Metrics
(using Serial Replacement)

Size of Isolated Fault Group	Probability	Cumulative Probability
1	100.00%	100.00%

Upon reflection, it should not be surprising that the use of serial replacement allows all simulated failures to be corrected by a single replacement. Remember, because this simulation used no diagnostic delay, only a single item will have failed each time that the diagnostic strategy is invoked. The only time that the fault group containing two components (A & B) is isolated is when item A has failed. Furthermore, for this set of failure rates, item A is determined to be more likely to have failed when that fault group is isolated, so it is replaced first when the fault group is serially repaired. Because item A is replaced first, this isolated fault group will be repaired with a single replacement whenever it is assumed that only a single malfunction exists at the time that the system is diagnosed. (If item B, however, were to have been assigned a failure rate large enough that function B2

fails more frequently than A, failures to A may have required two serial replacements to correct).

Now, if we increase the time between applications of the diagnostics, we will increase the likelihood of there being multiple, simultaneous malfunctions as the system is diagnosed. The following table (Table 15) shows the effects that diagnostic delay has upon simulation-based Fault Resolution statistics (when serial replacement is in effect):

Table 15. Effects of Diagnostic Delay Upon Fault Resolution (using Serial Replacement)

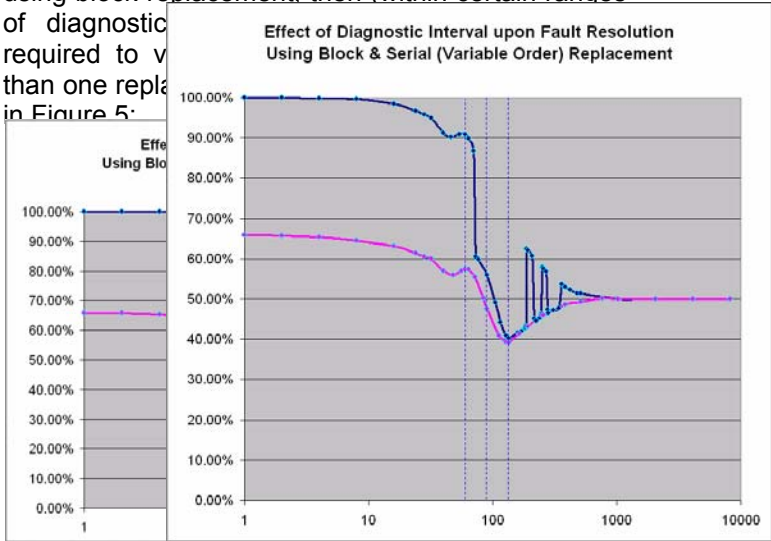
Length of Delay (in hours)	Likelihood of Repairing a Fault with 1 Replacement
0	100.00%
1	99.99%
2	99.97%
4	99.89%
8	99.58%
16	98.39%
32	95.05%
64	89.86%
128	69.18%
256	57.59%
512	51.42%
1024	50.03%
2048	50.00%

As would be expected, the likelihood of repairing the system with a single replacement decreases as the interval between diagnostics grows. When diagnostics are performed immediately after each fault, failures can always be repaired with a single replacement. At the other extreme, if diagnoses are separated by a long enough time interval, then all functions will have had the chance to fail by the time that the diagnostics are invoked (this is, of course, purely theoretical since the system would in all likelihood stop being used once certain functions have failed). Nevertheless, if all functions were to malfunction as the design is tested, the first isolated fault group would contain functions A and B2, thereby requiring two replacements to repair. The diagnostics would next isolate item C which, once replaced, would fully repair the system. When there is a large delay between diagnostic sessions, failures are observably repaired with two replacements (A & B) half of the time and with one replacement (item C) the other half. So, when this system is maintained using a multiple-fault isolation strategy and a fixed serial replacement order, the likelihood of visibly repairing a fault with a single maintenance action varies from 100% to 50%, depending upon the interval between diagnostic sessions. A similar trend (Table 16) can be observed when the system is maintained using block replacement:

Table 16. Effects of Diagnostic Delay Upon Fault Resolution (using Block Replacement)

Length of Delay (in hours)	Likelihood of Repairing a Fault with 1 Replacement
0	66.12%
1	65.95%
2	65.76%
4	65.43%
8	64.54%
16	63.05%
32	59.93%
64	57.33%
128	39.33%
256	46.06%
512	49.27%
1024	49.98%
2048	50.00%

As is the case when the design is maintained using serial replacement, the likelihood of repairing a fault by replacing a single item converges on 50% as the delay between diagnostic sessions increases. If block replacement is used, however, the curve approaches the limit (50%) from below, whereas with serial replacement, the limit is approached from above. This means that, if the design is maintained using block replacement, then (within certain ranges of diagnostic
required to v
than one repla
in Figure 5:



In this graph, the decrease in Fault Resolution as diagnostic delay increases has been mapped onto a logarithmic time axis. The higher curve depicts the likelihood of observably repairing a malfunction with a single replacement action when the design is maintained using serial replacement. The lower curve, on the other hand, represents the likelihood of repairing a malfunction with a single replacement when block replacement is employed. The dotted vertical lines depict where the three component MTBFs (60, 90 & 135 hours) fall on the time axis. Predictably, these lines intersect the curves at points where the slope changes. After the final MTBF (135), the relatively slow convergence upon 50% is due to the fact that failures to that component (C) fall into an exponential distribution.

The serial replacement algorithm that we have used in these simulations has assumed that items are replaced in accordance with a fixed replacement order based on the relative likelihood of failures when there is no diagnostic delay. Now let's look at the curves that result if the serial replacement order were to be optimized for each simulation, taking into account the effect of diagnostic delay (Figure 6).

As would be expected the lower curve is no different from before (since block replacement is in no way affected by this change). The serial replacement curve, on the other hand, displays some curious and perhaps unexpected discontinuities. Although the curve converges upon 50% at the same rate as it did when a fixed replacement order was used, it now alternates between converging from above (as it did before) and below (such as when block replacement is utilized). In effect, the Fault Resolution jumps between two curves as it converges toward 50%.

Figure 6. Effect of Diagnostic Interval upon Fault Resolution (Variable Replacement Order)

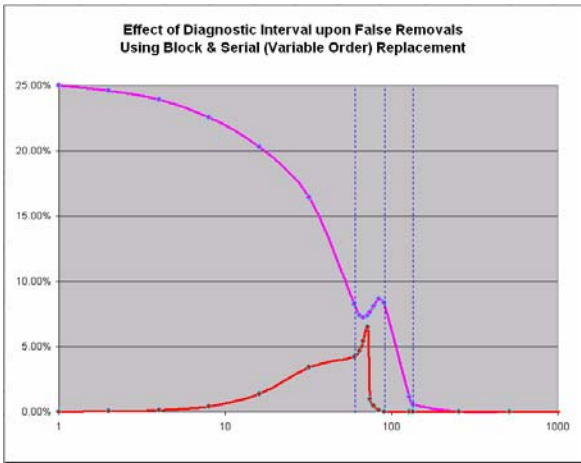
The reason for this discontinuity lies in the use of two different replacement orders for the fault group containing functions A & B2. Put succinctly, in a multiple-failure scenario, if item A is serially replaced before item B, the curve converges from above (a fault is *more* likely to be observably repaired with a single replacement); conversely, if B is replaced before A, the curve converges from below (a fault is *less* likely to be observably repaired with a single replacement). To understand why this is so, we must first consider the functional topology of our example design. Figure 7 shows the functional block diagram that corresponds to the design depicted in Figure 4:

Figure 7. Functional Block Diagram

For this design, serial replacement is only employed when the diagnostics isolate to the fault group containing functions A and B2. If item A is replaced first, then the only time that the effect of item A's replacement would *not* be observable would be when functions B1, B2 & C present multiple, simultaneous malfunctions as the system is diagnosed (this is rare, unless the diagnostic interval is relatively long). On the other hand, if item B were to be replaced before item A, then there would be no observable effect any time that A is malfunctioning as the system is diagnosed (since item A is upstream from all of the system's test points). Since the likelihood of function A failing is considerably higher than the likelihood of functions B1, B2 & C simultaneously malfunctioning, the likelihood of resolving a failure with a single item replacement is substantially worse when item B is replaced before item A. Because item B has a higher failure rate than item A, however, the diagnostic simulation determines that B should be replaced first when that fault group is isolated (that is, when the interval between diagnostic sessions is not taken into consideration). This means that a more optimized replacement order can produce a less attractive Fault Resolution number.

All this, of course, begs the following question: if it cannot improve the Fault Resolution of a system, why take diagnostic delay into consideration when optimizing the serial replacement order? The answer to this question is quite simple, if not immediately apparent: if the serial replacement order is optimized based on knowledge of what failure combinations are likely at the time that diagnostics are performed, then the expected number of false removals will be reduced. Because it takes the diagnostic interval into account as it determines the replacement order, the diagnostic simulation is able to optimize (minimize) false removals—even though, as we shall discuss shortly, this optimization may come at the expense of Fault Resolution.

Figures 8 and 9 depict the changes in the false removal rate as the interval between diagnostic



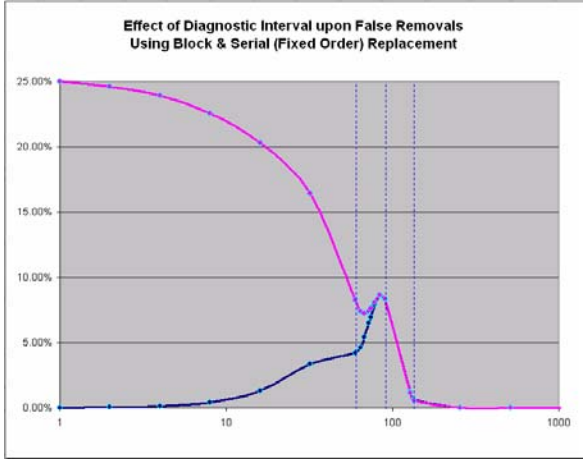
sessions increases. Both figures include the curve for the change in false removals when the system is maintained using block replacement (this is the curve that starts at the upper left-hand corner of each graph). The second curve in Figure 8 shows the change in the false removal rate when the design is maintained serially using a fixed replacement order. In Figure 9, on the other hand, the second curve shows the change in false removals (as the diagnostic interval increases) when the system is maintained using a serial replacement order that has been optimized to take diagnostic delay into account.

As would be expected, the use of serial replacement results in fewer false removals than if the system were maintained using block replacement. If a fixed replacement order is used (prioritized, once again, by the relatively likelihood of a failure to each function resulting in the given fault group being isolated), then the improvement in false removals is experienced most when diagnostic sessions are separated by shorter time intervals. As the interval between diagnostic sessions increases, there is a reduction in the profit to be gained from employing a serial replacement strategy. Eventually—in this example, when the delay between diagnostics approaches 80 hours—the effect of serial replacement upon the false removal rate is no different from when block replacement is employed.

Figure 8. Effect of Diagnostic Interval upon False Removals (Fixed Replacement Order)

Figure 9. Effect of Diagnostic Interval upon False Removals (Variable Replacement Order)

If the system is maintained serially using a variable (optimized for each diagnostic interval) replacement order, then the false removal rate not only remains less than the rate that would be achieved using block replacement; it also approaches zero more quickly. In other words, using this maintenance philosophy, zero false removals can be attained with smaller intervals between diagnostic sessions than would be needed using block replacement. The curve starts out the same as for fixed (un-optimized) serial replacement. When the diagnostic delay is slightly longer than 70 hours, however, the false removal rate suddenly dives from about 6.5% to about 1% and quickly drops to zero. This drop corresponds, incidentally, to a large drop in Fault Resolution over the same range (Figure 6)—a drop that did not occur when the system is serially maintained using a fixed replacement order (Figures 5 & 8). Interestingly, the sharp discontinuities that



appear in Figure 6 as the Fault Resolution converges upon 50% all occur after the false removal rate has already settled on 0.0%. In short, the same optimization that allows the false removals to drop sharply for diagnostic intervals between 60 and 90 hours also produces relatively poor Fault Resolution statistics for some diagnostic intervals in excess of 188 hours.

This disjunction between the False Removal Rate and Fault Resolution metrics is due neither to the accuracy of the simulation nor the specific serial replacement order that is employed (although this disjunction is glaringly apparent for time-optimized replacement orders). Instead, the reason why improvements in the False Removal Rate can result in less attractive Fault Resolution numbers lies in the definition of the two metrics. The Fault Resolution metric quantifies the replacements needed to visibly repair a malfunction; the False Removal Rate, on the other hand, quantifies the replacements that actually repair a malfunction, regardless of whether the existence of multiple failures masks the technician's ability to observe the corrected functionality. What this also means is that, even when calculated using a diagnostic simulation, Fault Resolution numbers do not consistently reflect the ability of a diagnostic (and replacement) strategy to reduce the number of unnecessary maintenance actions.

What is needed is not a better Fault Resolution metric—Fault Resolution remains the best measure of diagnostic performance that can be verified directly using field data. Rather, what is needed is a new metric that quantifies the likelihood of correcting a component malfunction with a single replacement (as opposed to the likelihood of observably correcting a failure with a single replacement). This new metric would then vary predictably in accordance with the False Removal Rate.

Example 3: False Removals

As we have already seen, a diagnostic simulation—particularly when equipped with a multiple-failure fault isolation strategy and a maintenance engine

that can perform block and/or serial maintenance—is able to simulate the detection, isolation and replacement of any combination of malfunctioning items. Moreover, since a simulation knows the precise combinations of functional failures that result in each fault isolation, it can estimate the effect of diagnostic ambiguity upon the false removal rate. It is important to recognize, however, that diagnostic simulations cannot predict actual false removal rates—these rates can only be calculated from support data and may be influenced by many factors in addition to diagnostic ambiguity. Nevertheless, since ambiguity is a prime cause of false removals, it is useful to be able to estimate the false removals that can be attributed to sub-optimal diagnostics. As we shall see, these statistics may be used not only as baseline numbers for the prediction of actual false removal rates, but also (all other things being equal) as a measure of how maintenance requirements might change over the intended lifetime of a system or equipment.

Throughout this example, the following design (Figure 10) shall be used:

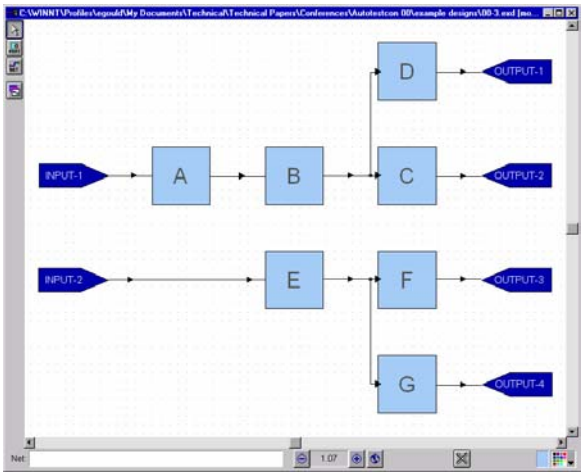


Figure 10. Design for Example 3

The MTBFs and Replacement Costs for the seven components in the design are listed in Table 17 (for this example, it shall be assumed that all failures for each item fall into a standard normal distribution). Furthermore, each of the items has only one function. Diagnostics are assumed to be performed at six hour intervals.

Table 17. Assigned MTBFs and Replacement Costs

Item	MTBF	Replacement Cost (in U.S. dollars)
A	1.5 years	20.00
B	250 days	2.00
C	1 year	90.00
D	2 years	350.00
E	4.5 years	200.00
F	3.75 years	10.00
G	3.75 years	25.00

Each of the false removal statistics for this example will be based on 1000 simulations over the specified mission length. Let's first look at false removals when the system is maintained for one year using block replacement. Over the first year, nearly 62% of all replaced items are falsely removed—that is, replaced prior to their having failed. Worse yet, for two of the components (A & C), over 90% of all replacements during the first year are premature.

The diagnostic simulation allows us to compare the premature replacements for each component with the replacements that would have resulted if the item were only replaced when it had failed. From this comparison, we can compute the average percentage of each component's lifetime that remains when it is prematurely replaced, as well as the average number of extra replacements (replacements beyond those that would have occurred if each component were only to be removed when it had failed). The resulting numbers have been compiled into Table 18:

Table 18. False Removal Statistics for the First Year of a System Maintained using Block Replacement

Item	False Removal Rate	Premature	Extra Replace-ments	Cost due to Extra Replace-ments
A	94.80	50.01	0.93	18.60
B	1.70	0.39	< 0.01	0.02
C	90.18	34.43	0.64	57.60
D	0.00	N/A	N/A	N/A
E	0.00	N/A	N/A	N/A
F	0.00	N/A	N/A	N/A
G	0.00	N/A	N/A	N/A

The **False Removal Rate** column contains the percentage of all replacements of the given item that occurred prior to that item actually having failed.

The **Premature** column lists the average percentage of the lifetime of the given item that remained when

that item was replaced. Item A, for example, is being replaced while half of its useful life remains.

Extra Replacements refers to the average number of replacements beyond those that would be expected if the item were to only be replaced when it fails.

Cost due to Extra Replacements refers to the cumulative item Replacement Cost associated with all extra replacements. This number represents the contribution of diagnostic ambiguity toward the expected cost to maintain the system. As can be easily seen, the cost due to extra replacements is negligible during the first year of deployment.

Let's compare these numbers with the False Removal statistics that would result if the mission length for our system were to increase to 30 years (Table 19):

Table 19. False Removal Statistics for Thirty Years of a System Maintained using Block Replacement

Item	False Removal Rate	Premature	Extra Replace-ments	Cost due to Extra Replace-ments
A	97.73	51.31	23.64	472.80
B	2.27	0.40	0.20	0.40
C	87.64	37.11	19.88	1789.20
D	0.00	N/A	N/A	N/A
E	0.02	4.27	< 0.01	< 2.00
F	0.00	N/A	N/A	N/A
G	56.71	28.55	3.34	83.50

A quick comparison of Tables 18 & 19 shows that, in addition to extending the period over which items A, B & C can be falsely removed, the longer mission length presents items E & G with the opportunity to be falsely removed as well. For item E, however, false removals result in extra replacements in fewer than one mission out of every hundred.

Now we'll look at the results if this system were maintained using serial replacement.

The following table (Table 20) lists the false removals that would be expected if this system were to be maintained using serial replacement (with a variable replacement order):

Table 20. False Removal Statistics for the First Year of a System Maintained using Serial Replacement

Item	False Removal Rate	Premature	Extra Replace-ments	Cost due to Extra Replace-ments
A	0.00	N/A	N/A	N/A
B	3.92	1.72	0.05	0.10
C	0.00	N/A	N/A	N/A
D	0.00	N/A	N/A	N/A
E	0.00	N/A	N/A	N/A
F	0.00	N/A	N/A	N/A
G	0.00	N/A	N/A	N/A

Although nearly four percent of all replacements of item B constitute false removals, this premature replacement will only occur in 5 out of every 100 replacement actions (the average cost being 10 cents per design). Table 21 depicts the statistics that would result from a thirty year simulation utilizing serial replacement.

Table 21. False Removal Statistics for Thirty Years of a System Maintained using Serial Replacement

Item	False Removal Rate	Premature	Extra Replace-ments	Cost due to Extra Replace-ments
A	0.05	0.02	< 0.01	0.20
B	36.34	18.06	10.18	20.36
C	0.00	N/A	N/A	N/A
D	0.00	N/A	N/A	N/A
E	0.01	0.01	< 0.01	0.20
F	0.00	N/A	N/A	N/A
G	0.00	N/A	N/A	N/A

If one were to base long-term maintenance decisions on short-term data, then one might assume that the false removal rate for item B in the first simulation remains constant over time. On the contrary, item B's false removal rate increases over time—from 3.92% to 36.34% over the course of thirty years.

Conclusions

Simulation-based Fault Resolution and False Removal metrics provide greater accuracy than do traditional approaches to calculating the metrics. Metrics calculated using a Monte Carlo simulation

are free of many of the biases that inhere in traditionally-calculated metrics. Furthermore, because these diagnostic simulations can take into consideration a large number of different criteria, they can be used to produce more complex sets of data than can be done using traditional methods of assessing diagnosability. Ultimately, because diagnostic simulations are even able to trace events that have never occurred, complex and useful new metrics (such as the effect of diagnostic ambiguity upon Life Cycle Cost) can be easily derived from simulation data.

References

[1] "Appendix A." *Testability Program for Electronic Systems and Equipments*, MIL-STD-2165, 1985, pp. 55–56.

[2] "Procedure V." *Maintainability Prediction*, MIL-HDBK-472, 1966 (Revised 1984), pp. V-6–V-7.