Interval Methods for Fault-Tree Analysis in Robotics

Carlos Carreras, *Member, IEEE*, and Ian D. Walker, *Member, IEEE*

***Abstract*—This paper describes a novel technique, based on interval methods, for estimating reliability using fault trees. The approach encodes inherent uncertainty in the input data by modeling these data in terms of intervals. Appropriate interval arithmetic is then used to propagate the data through standard fault trees to generate output distributions which reflect the uncertainty in the input data. Through a canonical example of reliability estimation for a robot manipulator system, we show how the use of this novel interval method appreciably improves the accuracy of reliability estimates over existing approaches to the problem of uncertain input data. This method avoids the key problem of loss of uncertainty inherent in some approaches when applied to noncoherent systems. It is further shown that the method has advantages over approaches based on partial simulation of the input-data space because it can provide guaranteed bounds for the estimates in reasonable times.**

***Index Terms*—Fault tree, interval method, reliability estimation, robotics.**

Acronyms [1]

FrPr	failure probability
FT	fault tree
MCS	Monte Carlo sampling
LHS	Latin hypercube sampling
s-	implies the statistical meaning

Notation

S_i	Robot sensor # i, optical Encoder, with FrPr s_i
M_i	Robot actuator # i, Electric Motor, with FrPr m_i
J_i	Robot joint # i, with FrPr j_i
λ	component failure rate
t	mission time
p	component FrPr
u, v, w, x	p interval points

I. INTRODUCTION

IN THE PAST few years, a key driver for developing and applying new technologies in robotics has been the need to send robots into hazardous & remote environments (*eg*, space, nuclear, undersea), where a failure in the robot can have disastrous consequences, and where repair is often impossible. This has motivated intensive research in fault tolerance (fault detection & isolation) and reliability (modeling and analysis) of robots [30], the latter being the topic of this paper.

The primary tool for reliability analysis in robotics has been FT analysis [30]. FT [9], [29] support the encoding of fault information from widely disparate subsystems into a single formal framework. With strong motivation & funding coming from the nuclear & hazardous waste clean-up communities in both the USA [26] and Europe [15], numerous FT studies have been performed over the past few years to identify & analyze key failure paths within different types of robots [12], [15], [30]–[32]. These studies have almost exclusively been limited to qualitative analyzes (*eg*, cut-set generation, identification of uncovered failure modes) due, in part, to the difficulty of obtaining good input probability data (from mean time to failure or failure rates of subsystems and components) for quantitative analysis, as robots of interest are typically 'one of a kind', or 'several of a kind' devices. Thus quantitative estimates have been largely untrusted, which in turn has led to overconservative designs, or even the nondeployment of some systems [32]!

The problem of uncertainty in low-level input FT data affecting the accuracy of (and hence assurance in) output reliability estimates from the FT has been recognized for some time. The structure of the FT itself has been used by several authors to synthesize attempts to combat this problem. In [20], the standard deviations of the reliability statistics were used to bound the standard deviation of the top-event probability in FT analyzes. A combination of exact expressions for, and bounds on the values of, the variance of the necessary combinations of events is used to propagate the standard deviations through the FT [20]. However, this analytic approach implies specific knowledge of the closed-form standard deviations of the input data (primary events). This is typically unavailable for many engineering applications such as robotics, and our motivation for this work was to find a method to represent more general uncertainty.

As an attempt to deal with more uncertain situations, the use of Dempster-Shafer theory to model imprecise input data is proposed in [11]. Dempster's rule of combination (as opposed to Bayes rule) is used in [11] to assign *masses* (as opposed to probabilities) to input events and gate logic in a FT. This allows user uncertainty in both the FT logic and the appropriate assignment of probabilities given trial data to be reflected in the calculations. The resulting output information is expressed in terms of the three logical variables: True, False, and Uncommitted (*i.e.*, a logical representation of uncertainty). However, in many applications, users would like an output representation that contains more information about the structure of the uncertainty.

Manuscript received October 5, 1999; revised November 14, 2000. Carreras is supported by projects ESPRIT 24137 and CICYT TIC97–0928. I.D. Walker is supported by the US National Science Foundation (NSF) under grant CMS-9796328, by NSF/EPSCoR grant EPS-9630167, and by the US Department of Energy (through Foster-Miller Inc.) under contract #DE-FG07–97ER14830, subcontract #805-05002.

C. Carreras is with the Departamento de Ingeniería Electrónica E.T.S.I. Telecomunicación, Universidad Politécnica de Madrid, 28040 Madrid, Spain (e-mail: Carreras@die.upm.es).

I. D. Walker is with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634 USA (e-mail: IanW@ces.clemson.edu).

Publisher Item Identifier S 0018-9529(01)06816-6.

[1] The singular & plural of an acronym are always spelled the same.

Another approach to dealing with uncertainty that has been popular recently is that of fuzzy logic. Accordingly, several authors [10], [24], [27], [28] have advocated 'fuzzifying' FT input data. For example, the idea of fuzzy data has been applied to robot reliability in [32], where each input was treated as a range (essentially the range of 'unit possibility' of a fuzzy membership function reflecting the uncertainty of the input data), resulting in a corresponding range for the output reliability estimate. However, as shown in [16], this approach can lead to extremely conservative results, and can result in loss of uncertainty in some cases, as explained in Section II.

Commercial risk assessment computer programs performing uncertainty analysis are usually based on Monte Carlo simulation [21], [25]. This method collects statistical results from the processing of random samples of the input data space. Its mathematical basis establishes essentially that (after a typical implementation), if well-characterized input *s*-normal distributions are considered and N output values are obtained from the simulation of N input vectors of random samples (output mean μ and standard deviation σ), it can be assumed that the real output mean is inside the range $[\mu - 3\sigma, \mu + 3\sigma]$ with a probability 0.997, with an error given by $3\sigma/\sqrt{N}$. This error tends to 0 as N increases. However, when considering uncertainty analysis, input distributions are not really known, so they must be approximated with known distributions or described in terms of histograms to obtain estimates of the statistical parameters of the outputs. In general, the accuracy of these estimates improves with the number of simulated samples. But this is only an approximate relationship, because the increase is not strictly monotonic, and it is not known how many samples are required for a given accuracy. In Monte Carlo simulation, this problem is particularly important when input distributions are very different from *s* -normal distributions (*e.g.*, multimodal distributions that have several peaks), and when large input spaces are considered. In these situations, estimates based on Monte Carlo methods should be interpreted carefully. Another problem of this approach is that it is not guaranteed that the worst case scenario (*eg*, inputs causing the maximum FrPr of the top event) is actually simulated. Therefore, this method cannot be used to obtain bounds on the output probabilities.

Over the past few years, there has been increasing interest and activity in the theory and application of Interval Arithmetic and Interval Computation [1], [17]–[19]. These techniques arose from the need to deal with uncertainty (*eg*, manipulate imprecise data, keep track of round-off error). They have been used in numerous applications recently [2], [23] and constitute the formal basis of the uncertainty propagation approach in this paper.

We are thus motivated herein to represent input data in terms of intervals. The standard *interval* data type is extended with a 'mass' so intervals can be used to represent 'densities' in the form of histograms [3], [4]. Note that closed form densities or distributions are not required, and the data can come from standard reliability data. The method is based on formal interval computations [1], [17], [23] and represents histograms by using predefined interval models called *grids* which allow control of the tradeoff between the accuracy of the results and the complexity of the computation. One major advantage of this interval approach is that it is often possible to simulate the whole interval input space in reasonable times, so the results are safe, meaning that it is guaranteed that the results from the worst case scenario are included in the output histograms. This is a clear advantage over methods based on partial simulation of the input data space (*eg*, Monte Carlo), which cannot guarantee generation of the worst case without complete simulation of all possibilities, as opposed to all intervals (appreciably less computation) in our method.

The interval approach in this paper is based on the approach in [5]. However, the version in [5] is valid only for integer values, and the reliability application requires real numbers. The algorithm was modified to allow consideration of real numbers in [8], in which the structure of estimates from the fuzzy and interval approaches were compared for the robot application for the first time. The case of loss of uncertainty using the fuzzy approach, and the inherent ability of the interval approach to avoid this problem was discussed in [7]. However, the interval estimates in [7], [8] are obtained in 'brute-force' fashion, requiring extensive computation, and in some cases forcing restricted analysis of the problem. This paper introduces a new generalized and fully parameterized version of the method, which allows more complete analysis. Its distinctive feature is the new set of grid models that allow improved efficiency of the computation process.

In Section II, an illustrative example of a 3-joint robot with redundant sensors is presented and evaluated in terms of some current approaches (fuzzy, Monte Carlo). Then, the generalized interval method is formally introduced. Results are obtained showing that the interval method improves the accuracy (measured in terms of (1) in Section IV) of the reliability estimates in robot FT analyzes over those in [32], avoiding the loss of uncertainty detected when considering the aging of the system (*eg*, time-varying reliability). This conclusion (discussed further in Section II) was predicted in [7] using the *rare event* approximation $\Pr\{A \cup B\} = \Pr\{A\} + \Pr\{B\}$ when computing the FT, and it is confirmed here with results obtained from the exact FT computation accompanied with detailed statistics of the process. It is also shown that the interval method not only appreciably improves the efficiency of algorithms based on exhaustive simulation of the input space [16], but constitutes a feasible alternative to Monte Carlo simulation in terms of computation times, with the advantage of providing guaranteed bounds for the results.

II. Illustrative Example

Assumptions:

1) The rare-event approximation, $\Pr\{A \cup B\} = \Pr\{A\} + \Pr\{B\}$, is accurate.
2) All failure events are mutually *s*-independent.

This approach uses the example of a planar robot manipulator, in Fig. 1. This type of robot structure is analogous to the 'upper arm–forearm – hand' structure of the human arm, when operated in the plane. The rigid links are connected by 3 rotational joints, each driven by its own electric motor. Positional feedback is obtained by using dual (redundant) optical encoders at each joint. This arrangement is fairly typical for robot manipulator subassemblies in remote environments [32].

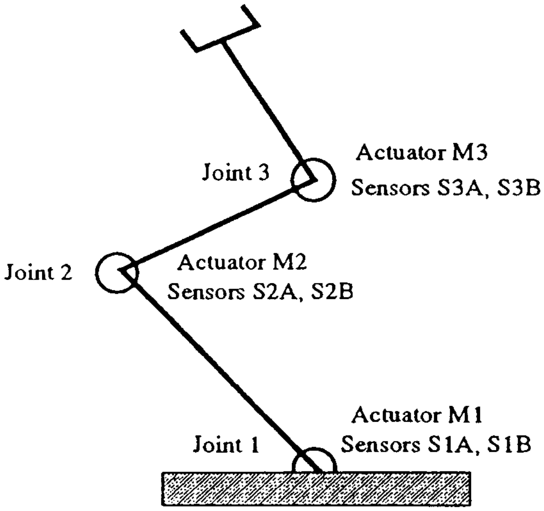

Fig. 1. 3-joint planar robot with redundant sensors.

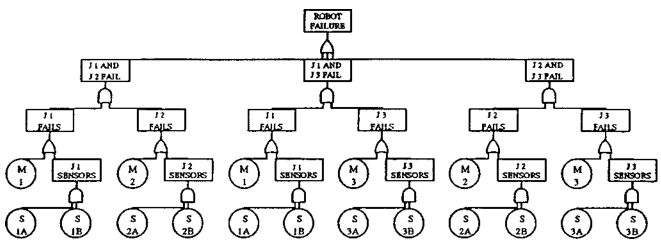

Fig. 2. FT for 3-joint planar robot with redundant sensors.

The task is to position the end-effector arbitrarily in its planar workspace. Since this can be achieved with any 2 of the joints, the robot has a redundant degree of freedom. For this robot, failure therefore occurs when any 2 of the joints are rendered immobile (joints are considered locked by brakes upon failure—the issue of failures in the brakes themselves is beyond the scope of this example). In turn, each joint fails if either its motor fails, or both of its encoders fail.

This logic can be easily encoded into a FT for the robot [32], with top event 'failure of the robot', as in Fig. 2. This example, first introduced in [32], compares & contrasts our results with those in [32], where the concept of propagating 'FrPr ranges' (rather than fixed values for the probabilities of subsystem failures) through the FT was proposed. This concept is consistent with the notion of fuzzy FT analysis introduced in [28], in which the input data are represented by a 'fuzzy trapezoid' describing the *possibility* of the input FrPr, as shown in Fig. 3. Note that this FrPr has a possibility of 1 for $v \leq p \leq w$. Between u and v, and between w and x, the possibility varies linearly between 0 and 1. Thus the set (u, v, w, x) uniquely describes each distribution.

The idea of using intervals, or 'fuzzy probability ranges', is appealing since this often fits the available data. For example, existing reliability data for electric motors (typically used in robot manipulators) across a wide range of applications reports an overall FrPr of 0.009 24, together with information that 68% of cases will be between 0.22 and 4.5 times this value, and 90% of cases will be between 0.08 and 1.8 of it [32]. Thus, for this case, (u, v, w, x) are estimated as

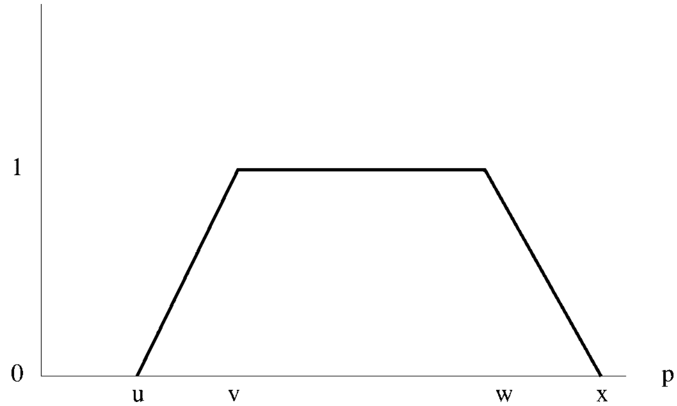

Fig. 3. Representation of input data as fuzzy possibility.

(0.000 739 2, 0.002 03, 0.041 58, 0.109 032) so that 90% of the cases are represented under the 'possibility distribution'. For a typical optical encoder, a corresponding representation of (0.001 24,0.003 41,0.0698,0.184) is reported in [32].

These 'fuzzy estimates' can be summed and multiplied using the rules [28]:

$$\text{Addition: } (u, v, w, x)_{\text{new}} = \big(u_1 + u_2, v_1 + v_2, w_1 + w_2, x_1 + x_2\big);$$

$$\text{Multiplication: } (u, v, w, x)_{\text{new}} = \big(u_1 \cdot u_2, v_1 \cdot v_2, w_1 \cdot w_2, x_1 \cdot x_2\big);$$

where the input distributions are: (u_1, v_1, w_1, x_1) and (u_2, v_2, w_2, x_2).

The use of these rules allow the combination of such fuzzy estimates through the usual FT logic gates, (AND, OR, *etc.*), while preserving the essential fuzzy structure (these estimates above are computed pairwise from the corners of the input trapezoids). That method, which we term the 'fuzzy approach', was used in [32] to evaluate the effects of uncertainty on the reliability of the encoders & motors on the overall robot. For example, given the numerical distributions for electric motor and optical encoders given in this section, and considering the event E, 'joint J_i fails', its probability is $s_{i,A} \cdot s_{i,B} + m_i$ (using assumption #1). Therefore, since input probabilities are for *s* -independent events (assumption #2), the output trapezoid for E is:$E(u, v, x, w) = (0.000\,740\,7, 0.002\,04, 0.046\,45, 0.142\,89)$.

This result reflects the uncertainty in the output, beyond the uncertainty which is obtained from the traditional single value obtained via a quantitative FT analysis [32]. However, the 'fuzzy' operations used to generate this fuzzy trapezoid produce a very conservative estimate. Furthermore, such fuzzy manipulations have a severe limitation that was not unveiled until input probabilities varying with time (taking into account the aging of the system) were considered. This case is reported in [16] where it is shown that the fuzzy approach in [28], [32] and outlined here led to erroneous reliability range estimates of zero width. As defined here, this means complete loss of uncertainty!

For this robot example, consider the event F: Exactly 1 joint fails. From Fig. 2, (using assumption #2),

$$\Pr\{F\} = 3 \cdot (j_1 \cdot j_2 \cdot j_3) - 2 \cdot (j_1 \cdot j_2 + j_1 \cdot j_3 + j_2 \cdot j_3) + j_1 + j_2 + j_3;$$

$j_i \equiv$ probability of failure of joint $i, i = 1, 2, 3$, which are computed here as (no rare-event approximation),

$$s_{i,A} \cdot s_{i,B} + m_i - s_{i,A} \cdot s_{i,B} \cdot m_i.$$

Consider the fuzzy approach of [28], [32] and the failure rates specified for this example for the time-varying case (using the exponential distribution for component failure probability, $p(t) = 1 - \exp(-\lambda t)$, $\lambda \equiv$ the failure rates in the distributions[29]. Then at time $t = 18.55$, $v = w = 0.112$ (corresponding to Fig. 3). Thus the trapezoid becomes a triangle, or equivalently, the 'possibility range' between v and w vanishes! This exposes a serious limitation with the fuzzy approach, in which the uncertainty present in the physical situation is not reflected in the calculation. Clearly a more accurate approach is called for.

This loss of measure of the uncertainty is due to the way the ranges are combined in the fuzzy addition and multiplication. Only the endpoints of the ranges are combined, and then only in a pairwise fashion. However, best & worst case conditions at the endpoints (the shape of a trapezoid) are only preserved if *s*-coherent systems are considered

$$\frac{\partial \Pr\{p_j\}_{j=1,\ldots,N}}{\partial p_i} \geq 0, \quad \text{for all } i = 1, \ldots, N$$

where $\Pr\{p_j\}_{j=1,\ldots,N} \equiv$ the system probability, $p_i \equiv$ the input probabilities. Since f and j_i correspond to noncoherent systems, the fuzzy approach can provide erroneous estimates, depending on the input data, *eg*, data after the aging process.

The proposed solution [16] to this problem was brute-force exhaustive sampling of combinations of points within the ranges. However, this was noted to be computationally inefficient (in our example, 9 input independent input probabilities, described with 5 fractional digits yield an input space of 10^{45} possible combinations of input values); in any case, it left unclear how fine the sampling had to be to be effective.

Methods based on Monte Carlo simulation can be considered for evaluating the FT of the example in this section. In this case, a transformation of the available numeric data into well-characterized distributions or into histograms (as explained in Section III) is first required. The histogram representation is preferred here due to the uncertainty implied in the available data. As for the sampling of the huge input space in the previous paragraph, two major choices are provided in existing risk-assessment packages such as [13], [25]: MCS and LHS. For MCS, random numbers, usually uniformly distributed, are generated and then transformed to the histogram distribution. For large input spaces, MCS is often time consuming due to the many samples typically required [33], because distributions obtained from simulating a very small fraction of the input space can hardly be trusted [14].

LHS was developed to reduce the number of such samples [25], [33]. LHS:

- divides the range of each of the k input variables into n nonoverlapping intervals,
- randomly selects n values (one value from each interval) for each of the k variables,
- combines them randomly into n k-tuples which are used as input vectors.

While LHS reduces the number of samples, they are much harder to compute, and, in the words used in [22], [33], LHS becomes particularly useful only when an *exceedingly* sparse N-dimensional space must be sampled at M points. Consequently, both methods are regarded as time consuming when many samples must be generated, as in the example in this paper. In addition, there is no guarantee that the selected samples include the worst-case scenario, thus no safe output bounds can be derived from these methods (bounds can be estimated from the largest simulated results, but without any guarantee to be correct).

Section III reviews & discusses an alternative approach using interval methods; this approach avoids the problems detected in the fuzzy method while providing guaranteed output bounds, as opposed to MCS and LHS. Although LHS might resemble the interval method, it is quite different because the interval method is based on a different computation model with different data types (*viz*, intervals).

III. A Novel Interval Approach

The proposed new approach builds on our initial work [8], in which the input data are reformulated in terms of an estimate of the probability densities of the inputs in terms of histograms [3], [4]. From these histograms we compute the probability density of the event of interest, (top event of the FT, *etc*). For example, for the electric motor data in Section II, an input distribution (histogram) is constructed as follows:

$$\begin{aligned}
m =& (0, 0.000\,739) : 0.05\\
m =& (0.000\,739\, 0.002\,03) : 0.11\\
m =& (0.002\,03, 0.009\,24) : 0.34\\
m =& (0.009\,24, 0.041\,58) : 0.34\\
m =& (0.041\,58, 0.109\,03) : 0.11\\
m =& (0.109\,03, 1) : 0.05.
\end{aligned}$$

Thus m (representing the FrPr of an actuator M_i) has 0.05 of its probability mass between 0 and 0.000 739, *etc*. Fig. 4 is a plot of the corresponding histogram. Only the range [0,0.15] is included, in order to allow a more detailed representation. Note that in this type of distribution, in contrast to the fuzzy trapezoid, *all* the probability mass is included under the distribution. These histograms are described in terms of intervals which are defined as:

Definition: An interval $[a, b]/p$ is the set of $N = (b - a - 1)$ integers x that verify $a \leq x \leq b$ with an associated probability p.

Let p be uniformly distributed in $[a, b]$, so that the probability of any $x \in [a, b]$ can be computed as p/N. The interval definition #1 is based on integer numbers; thus some preprocessing and postprocessing are required for its application to the reliability problem where the data are based on rational numbers. In particular, the range [0,1] is partitioned into M steps of equal size represented by a number (midpoint) and a probability (obtained from the original distribution). Then, the [0,1] range is scaled into the range $[0, M - 1]$ and mapped to an integer distribution in the same range, where each integer n has the probability of the subrange $[n - 0.5, n + 0.5]$ in the scaled distribution. It should be noted that the scaling might require some

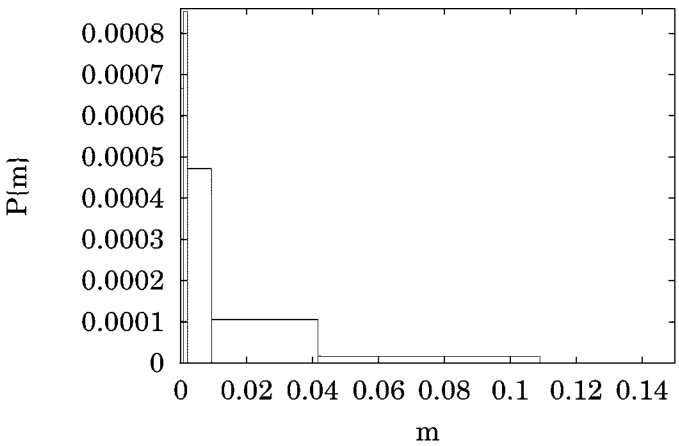

Fig. 4. FrPr of the Motor.

additional operations (multiplication or division by M) when computing the FT algorithm in order to maintain the original behavior. The value of M is important in the computation, because it determines the numeric accuracy. In general, M is selected according to the maximum precision provided in the input data so that no information is lost in the scaling process. For example, if the input data are provided with m fractional digits, a value $M = 10^m$ is used for the scaling.

The most distinctive feature of the interval approach is that it is heavily based on predefined interval models called grids, to represent the histograms through the computation [6]. Grids allow selecting the size of the data ranges (granularity) at the operation level, thus providing a mechanism to control the tradeoff between accuracy and computation-time. In general, the complexity of an interval-based computation is determined by the size of the interval input space and by the process of collecting the result of each run in a global output histogram. If unbounded interval models are used, then exact collection of results tends to increase the number of bars of the output histogram (merging 2 intervals with nonempty intersection produces 3 output intervals) and the collection time easily becomes prohibitive. Thus, bounded interval models based on grids are used. Grids are lists of disjoint adjacent intervals covering the whole numeric range of interest. In this situation, the collection process is governed by 2 rules:

> *Merge* rule: All interval results occurring inside the same interval of the grid are represented as a single interval with probabilities added.
>
> *Split* rule: Any interval result spanning several intervals of the grid is partitioned into as many intervals with proportional probabilities before being merged in the output histogram.

Several types of grids can be defined to control, not only the collection process, but the size of the interval input space (*ie*, grids are also applied to the input histograms).

Of course, the user can decide what histogram bars should be used to represent inputs and outputs [3]; in this case we use the term, custom grids. They are tedious to define and, to be effective, usually require some previous knowledge about the shape of the (output) distributions. In our interval approach, regular analytic grids are mainly considered. They are characterized by 4 parameters:

- type,
- granularity (g),
- center of symmetry (c),
- focus (f).

The type determines the analytic expression of the grid. The granularity represents the size of its intervals. The grid is symmetric with respect to c. The focus reduces the number of intervals of the grid to a finite number near c; this parameter is a distinctive feature from previous versions of grid models [5], [6]. It reduces the complexity of the computation by limiting the region of the numeric space which is analyzed in detail.

Two types of regular grids are considered:

- linear grids (formed by intervals of equal size),
- geometric grids (formed by intervals with exponentially increasing sizes as they depart from the center of symmetry).

Definition: A linear grid is a set of adjacent intervals $[A, B]$, each of them uniquely identified by integer n, that verify one of the identities:

$$[A,B] = \begin{cases} [-\infty, c + g\cdot(-f+1)], & n \le -f, \\ [c + g\cdot n + 1, c + g\cdot(n+1)], & -f < n < -1, \\ [c - g + 1, c - 1], & n = -1, \\ [c, c], & n = 0, \\ [c+1, c+g-1], & n = 1, \\ [c + g\cdot(n-1), c + g\cdot n - 1], & f > n > 1, \\ [c + g\cdot(f-1), \infty], & n \ge f. \end{cases}$$

Definition: A geometric grid is a set of adjacent intervals $[A, B]$, each of them uniquely identified by integer n, that verify one of the following identities:

$$[A,B] = \begin{cases} \left[-\infty, c - g^{(f-1)}\right], & n \le -f, \\ \left[c - g^{-n} + 1, c - g^{-(n+1)}\right], & -f < n0, \\ [c, c], & n = 0, \\ \left[c + g^{(n-1)}, c + g^{n} - 1\right], & f > n > 0, \\ \left[c + g^{(f-1)}, \infty\right], & n \ge f. \end{cases}$$

The g & f are positive integers.

If the focus feature is not used, the expressions in definitions #2 & #3 revert to those in [5] $(f \to \infty)$ in those expressions). It should be noted that the ability to set the center of symmetry to any point of the range is particularly important in the geometric grid, because the center is the point where the model is most accurate.

Once the input distributions have been transformed to integer ranges and represented in terms of a predefined grid, the computation method is adapted from [3] as:

1) Consider the input space $N_1 \times \cdots \times N_I$; N_i is the set of intervals describing the histogram of input i.
2) For each vector $(\ldots, [a_{i,j}, b_{i,j}]/p_{i,j}, \ldots)$ of the input space; $[a_{i,j}, b_{i,j}]/p_{i,j}$ represents bar #j of the histogram describing input i:
   a) Compute its probability $P \equiv \prod_{i=1}^{I} p_{i,j}$.
   b) Execute the operations using interval arithmetic.
   c) Assign P to each resulting interval.
   d) Collect the results in output histograms by applying the split and merge rules.

Standard definitions of operations between intervals are used for the computation [17], [23]. For positive intervals:

$$[x_1, y_1] + [x_2, y_2] = [x_1 + x_2, y_1 + y_2]$$
$$[x_1, y_1] - [x_2, y_2] = [x_1 - y_2, y_1 - x_2]$$
$$[x_1, y_1] \cdot [x_2, y_2] = [x_1 \cdot x_2, y_1 \cdot y_2]$$
$$\frac{[x_1, y_1]}{[x_2, y_2]} = \left[\frac{x_1}{y_2}, \frac{x_2}{y_1}\right].$$

It is clear that assuming uniform distributions in the output intervals is an approximation that should be considered in the analysis of the estimates.

Because the interval method computes the FT for all possible input interval combinations, it produces a much more complete estimate of the output distribution than the methods in [28], [32], where only the trapezoid corners are considered and in a pairwise fashion. In fact, noncoherent systems can now be evaluated, because interval computations guarantee that the interval result of an operation contains all the results that can be obtained when operating values from the input intervals. Therefore, even though best and worst conditions might not be represented by the interval endpoints, they are included in the histogram distribution collecting all the interval results, and the interval method can be used to obtain bounds for the output probabilities. If such bounds are the only result of interest, then the computation can be simplified to perform the Skelboe algorithm. This algorithm is based on the cyclic bisection of only the input intervals providing the largest output probabilities. Cyclic bisection provides tighter bounds on each iteration, thus there is no limitation on the accuracy that can be obtained. (For a detailed explanation see [17]).

Guaranteed bounds are a clear advantage of the interval method over Monte Carlo simulation and the method in [16]. It remains to be seen how they compare in terms of computation time. Our experiments show that, in general, interval computations do not add appreciable complexity. In fact, in several experiments the overhead caused by the collection process when using intervals can be comparable to the overhead caused by random sample generation in Monte Carlo (except when very detailed output grids are used). Thus, in this situation, each interval run of the FT takes approximately twice the computation as each Monte Carlo run.

The interval approach is particularly interesting for computing large input spaces dominated by large input ranges,*ie*, many digits in the input probability values. However, they are less effective when input spaces are dominated by many inputs. This is also a difficulty for Monte Carlo methods. The reference manual of [25] suggests considering *s*-correlation classes: sets of inputs with identical distributions which share the same sample in each run (heuristic). Another approach suggested in [5] is to partition the FT into *s*-independent (or quasi *s*-independent) subtrees with fewer inputs which are computed separately. For example, the FT of Section II can be obtained from the independent computation of the probabilities of events

- dual sensor $(S_{i,A}, S_{i,B})$ fails,
- joint J_i fails,
- robot arm fails,

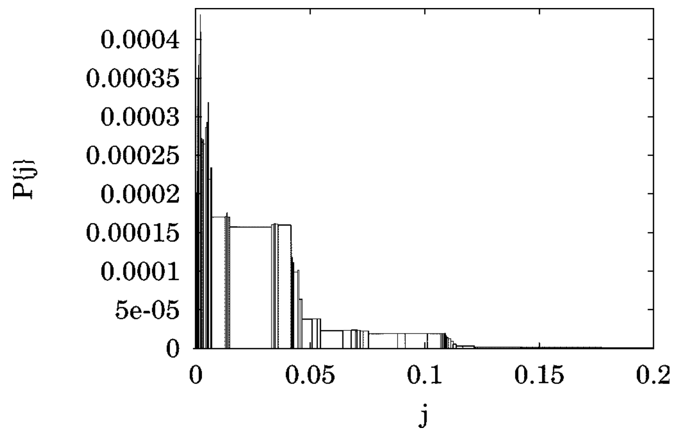

Fig. 5. Joint probability of failure (no grid).

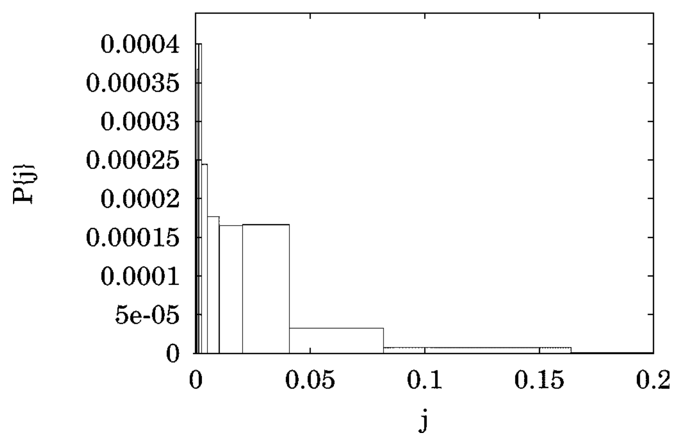

Fig. 6. Joint probability of failure (geometric input and output grids).

where each computation takes as input the output of the previous computation. The original 9 inputs are reduced to 2, 2, 3, respectively for each subtree. If subtrees are not truly *s*-independent, some oversize will appear in the interval results, but they will still be guaranteed bounds of the distribution.

IV. Results Using Interval Methods

The results of the interval method presented here are applied to the robot example in Section II. No rare event approximation is applied, as opposed to some initial results in [7]. The distributions obtained show much greater detail, and thus provide appreciably more information, than previous estimation results.

In a first set of experiments, the impact of using different types of grids during the estimation process is analyzed through the distribution obtained for E, the event 'joint J_i fails'. Fig. 5 represents the distribution obtained when using standard intervals (no grid). Fig. 6 uses geometric grids (grain 2, center 0, no focus) for both the input representation and the collection of results. Fig. 7 uses the same geometric input grid but the results are collected through a linear grid (grain 100, center 0, no focus). Grid parameters take into account that the input distributions are scaled to a range $[0, 10^5]$ (probability values given with 5 fractional digits). The output distributions are represented only in the range [0,0.2].

Table I compares some parameters of interest obtained from the simulations, including:

TABLE I
STATISTICS FROM THE SIMULATIONS FOR EVENT: JOINT J_i FAILS'

Input grid	Output grid	Interval operations	Output intervals	Output range	Prob. in [0,1]	Time (ms)
none	none	380	541	[-0.70695, 1.97994]	0.999453	1170
geo(2,0,-)	geo(2,0,-)	1024	404	[0,1.57053]	0.999746	330
geo(2,0,-)	lin(100,0,-)	1024	1958	[0,1.57053]	0.999746	2280
geo(2,0,-)	lin(100,0,0.2)	1024	487	[0,1.57053]	0.999746	390

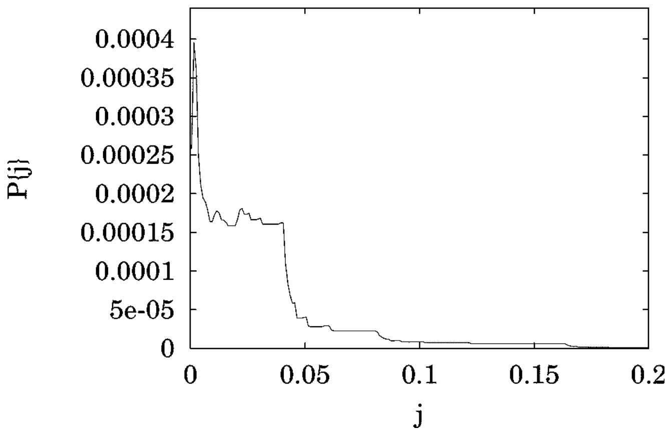

Fig. 7. Joint probability of failure (geometric input grid; linear output grid).

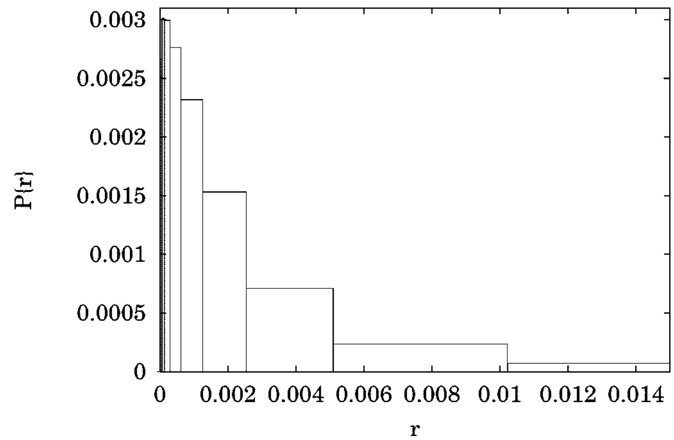

Fig. 8. Output distribution (probability estimate) of event 'robot-arm fails'.

1) the number of interval operations,
2) the number of output intervals, which gives an estimate of the memory requirements,
3) the output ranges which, in general, exceed the [0,1] range, due to the oversize implied by the interval computations,
4) the probability mass inside the [0,1] range,
5) the simulation time.

Table I also includes results when a focus value is used to limit the linear output grid to the range [0,0.2], thus reducing the collection time. The interval computations are performed in the integer domain by scaling the [0,1] range to a $[0, 10^5]$ range. Since interval operations only provide bounds for the output ranges with an accuracy determined by the data dependencies involved, it is possible that some results can fall outside the $[0, 10^5]$ range, thus leading to parameter #4 above. (Times should be taken only as relative measurements, because performance is not optimized in the current software implementation of the approach).

Considering a simulation strategy based on individual samples (Monte Carlo), the size of the input space is $3 \cdot 10^5$ input vectors that would require $15 \cdot 10^5$ operations for exhaustive computation. This number is greatly reduced in all interval-based simulations, thus proving their efficiency. When no grids are considered (Fig. 5) a great level of detail can be obtained. However, merging individual results in the output histogram increases substantially the simulation time, even for this simple case (more complex simulations show that this time quickly becomes too large). In addition, because the number of input intervals used for the computation is determined by the available data, the oversize of the output range is much greater than when using grids that enforce more input intervals; although in all cases the probability mass outside [0,1] is very small, showing little impact of the data dependencies on the accuracy of the interval results. Geometric grids (Fig. 6) allow fast computation while memory requirements remain low. A combination of a geometric input grid with a linear output grid (Fig. 7) maintains the computational complexity while providing greater detail in the results. Naturally, this implies an increase in the computation time due to the collection process which is mainly a function of the granularity of the grid. Therefore it can be adjusted, depending on the complexity of the computation, as opposed to the simulations without grids where no control mechanism is available.

The event R, 'robot arm fails' is analyzed in a second set of experiments. This is more complex; the FrPr is

$$r = j_1 \cdot j_2 + j_1 \cdot j_3 + j_2 \cdot j_3 - 2 \cdot (j_1 + j_2 + j_3);$$

$j_i \equiv$ failure-probability of joint J_i in Fig. 6. The results using geometric input and output grids with grain 2 and center 0 are in Fig. 8. Results with other grid combinations have similar features. The number of operations is 61 440 and the probability mass outside the [0,1] range still remains very low (0.999 857). The simulation time is 24 590 msec.

In this set of experiments, the interval-based results are compared to 'exact' results (in the integer domain) obtained from exhaustive computation of all individual integer inputs in order to evaluate the error introduced in the interval results. Exhaustive computation has been possible by using scaling factors N below 10^3 (probabilities with up to three fractional digits) and considering that all sensors and all motors behave identically (otherwise the size of the input space would have been too large). The error $\mathcal{E}$ is:

$$\mathcal{E} = \sum_{\forall i} \frac{|p_{i,\mathrm{ex}} - p_{i,\mathrm{int}}|}{2 p_{i,\mathrm{ex}}};$$

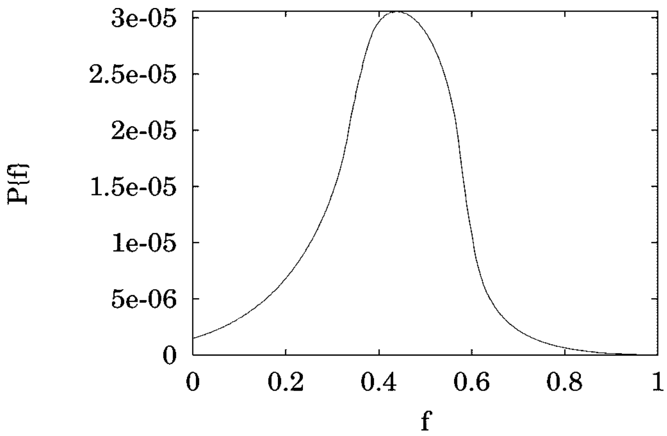

Fig. 9. Output distribution (probability estimate) of event 'exactly 1 joint fails' at time 18.55.

$\mathcal{E} \equiv$ the fraction of the exact output distribution, $p_{i,\mathrm{ex}}$, $p_{i,\mathrm{int}} \equiv$ probabilities of bar #i of the exact and interval-based output histograms, both represented in terms of the interval grid, the factor 2 accounts for the fact that each misplaced results causes a difference in the distributions of twice its probability. An error below 7% has been obtained in all comparisons, while computation times have been reduced as much as 4 orders of magnitude when using geometric grids. [8] also shows that, if computation times are fixed, then interval results provide smaller $\mathcal{E}$ than methods based on random sampling like Monte Carlo; *ie*, they are faster in getting closer to the true distributions. This is because the interval input vectors can be sorted so the that combinations with higher probabilities are computed first. In general, this is not feasible if individual numeric samples are selected.

The last set of simulations investigates the effect of aging input probabilities on system reliability. In particular, we are interested in the results provided by the interval approach when the fuzzy approach breaks down. Fig. 9 shows the output distribution for event F provided by the interval method using a linear input grid with grain 3000 and a linear output grid with grain 100. In this case, the joint failure probabilities used in the computation are previously obtained also using input and output linear grids. This provides much more detail at the cost of higher requirements in memory and computation-times. In particular, the simulation providing the distribution in Fig. 9 computes 589 560 interval operations in 1,600 seconds (including input generation and output collection times). The probability mass outside [0,1] still remains very low: 0.986 358. This result confirms that the fuzzy method, which shows a loss of uncertainty and peak at 0.112 for this example as explained in Section II, underestimates the true result.

In summary, these examples confirm that the interval methods in this paper result in more accurate & detailed estimates than with the fuzzy approach, and also produce consistent & meaningful results where the fuzzy method does not (loss of uncertainty). Also, by the appropriate use of grids, tradeoffs between accuracy and computational-complexity can be easily managed, as demonstrated in the examples. This allows comparable performance to Monte Carlo methods with the advantage that guaranteed distribution bounds are provided.

V. Applicability and Future Work

The interval-based reliability-estimation algorithm introduced in this paper has been illustrated using examples from robotics. We anticipate that the robotics arena, in which dealing with uncertain input data has proved a major obstacle in quantitative reliability estimation [32], is one area in which the method introduced here is likely to be particularly applicable. However, the interval approach to reliability estimation presented here applies to any system modeled by FT with input uncertainty represented by intervals. The ability of the approach to include formally the effects of the uncertainty in the input data in the output reliability estimate make the method very appealing across a range of applications. Our ongoing work focuses on the effects of different scalings & representations of the interval data on the output distributions.

References

[1] G. Alefeld and J. Herzberger, *Introduction to Interval Computations*, AP, 1983.

[2] (1995) Int'l Workshop on Applications of Interval Computations. [Online]. Available: http://cs.utep.edu/interval-comp/apictoc.html

[3] D. Berleant, "Automatically verified reasoning with both intervals and probability density functions," *Interval Computations*, no. 2, pp. 48–70, 1993.

[4] ——, "Bounding the results of arithmetic operations on random variables of unknown dependency using intervals," *Reliable Computing*, vol. 4, no. 2, pp. 147–165, 1998.

[5] C. Carreras, J. A. Lopez, and O. Nieto, "Bit-width selection in data-path implementations," in *Proc. 12th IEEE Int'l Symp. System Synthesis*, San Jose, Nov 1999, pp. 114–119.

[6] C. Carreras, O. Nieto, J. M. Moya, and F. Moya, "Fast data flow characterization," in *Proc. VIII BELSIGN Workshop*, Oct 1998.

[7] C. Carreras and I. D. Walker, "Interval methods for improved robot reliability estimation," in *Proc. 2000 Ann. Reliability and Maintainability Symp*, pp. 22–27.

[8] C. Carreras, I. D. Walker, O. Nieto, and J. R. Cavallaro, "Robot reliability estimation using interval methods," in *Proc. 1999 Workshop, Applications of Interval Analysis to Systems and Control*, Spain, pp. 371–385.

[9] J. B. Dugan and S. A. Doyle, "New results in fault-tree analysis," in *Tutorial Notes 1996 Ann. Reliability and Maintainability Symp.*, pp. 1–23.

[10] J. A. B. Geymayr and N. F. F. Ebecken, "Fault-tree analysis: A knowledge-engineering approach," *IEEE Trans. Reliability*, vol. 44, no. 1, pp. 37–45, Mar 1995.

[11] M. A. S. Guth, "A probabilistic foundation for vagueness and imprecision in fault-tree analysis," *IEEE Trans. Reliability*, vol. 40, no. 5, pp. 563–571, Dec 1991.

[12] B. M. Harpel, J. B. Dugan, I. D. Walker, and J. R. Cavallaro, "Analysis of robots for hazardous environments," in *Proc. 1997 Annual Reliability and Maintainability Symp*, pp. 111–116.

[13] Item Software, Av Sim [Online]. Available: http://www.itemsoft.com/Monte_Carlo.html

[14] J. Pesonen *et al.*, "Interval approach challenges Monte Carlo simulation," in *Proc. Scientific Computing, Computer Arithmetic, and Validated Numerics (SCAN-95)*, 1995.

[15] K. Lauridsen, P. Christensen, and H. E. Kongsø, "Assessment of the reliability of robotic systems for use in radiation environments," *Reliability Engineering and System Safety*, vol. 53, no. 3, pp. 265–276, 1996.

[16] M. L. Leuschen, I. D. Walker, and J. R. Cavallaro, "Robot reliability through fuzzy Markov models," in *Proc. 1998 Ann. Reliability and Maintainability Symp*, pp. 209–214.

[17] R. E. Moore, *Methods and applications of interval analysis*: SIAM, 1979.

[18] K. Nickel, *Interval Mathematics 1980*: Academic Press, 1990.

[19] ——, *Interval Mathematics 1985*: Springer-Verlag, 1996.

[20] L. B. Page and J. E. Perry, "Standard deviation as an alternative to fuzziness in fault tree models," *IEEE Trans. Reliability*, vol. 43, no. 3, pp. 402–407, Sep 1994.

[21] Palisade Corp. @RISK [Online]. Available: http://www.palisade.com/html/risk_for_project.html

[22] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in FORTRAN: The Art of Scientific Computing*: Cambridge Univ. Press, 1992.

[23] H. Ratschek and J. Rokne, *Computer Methods for the Range of Functions* Ellis-Horwood, Chichester, 1988.

[24] I. D. Roberts and A. E. Samuel, "The use of imprecise component reliability distributions in reliability calculations," *IEEE Trans. Reliability*, vol. 45, no. 1, pp. 141–144, Mar 1996.

[25] SAPHIRE, IRRAS [Online]. Available: http://saphire.inel.gov/saphire_main.html

[26] E. J. Shen, "Retrieval of underground storage tank wastes: The hanford challenge," in *Proc. 1995 American Nuclear Society Topical Mtg: Robotics and Remote Systems*, pp. 549–553.

[27] D. Singer, "A fuzzy set approach to fault tree and reliability analysis," *Fuzzy Sets and Systems*, vol. 37, no. 2, pp. 267–286, 1990.

[28] H. Tanaka, L. T. Fan, F. S. Lai, and K. Toguchi, "Fault-tree analysis by fuzzy probability," *IEEE Trans. Reliability*, vol. 32, no. 5, pp. 453–457, Dec 1983.

[29] W. E. Vesley, F. F. Goldberg, N. H. Roberts, and D. F. Haasi, *Fault Tree Handbook*: US Nuclear Regulatory Com, 1981.

[30] M. L. Visinsky, J. R. Cavallaro, and I. D. Walker, "Robot fault detection and fault tolerance: A survey," *Reliability Engineering and System Safety*, vol. 46, no. 2, pp. 139–158, 1994.

[31] I. D. Walker and J. R. Cavallaro, "Failure mode analysis for a hazardous waste clean-up manipulator," *Reliability Engineering and System Safety*, vol. 53, no. 3, pp. 277–290, 1996.

[32] ——, "The use of fault trees for the design of robots for hazardous environments," in *Proc. 1996 Ann. Reliability and Maintainability Symp*, pp. 229–235.

[33] C. N. Zeeb and P. J. Burns. (1998) A comparison of failure probability estimates by monte carlo sampling and latin hypercube sampling. Sandia Nat'l Labs; Dep't Mechanical Engineering Colorado State Univ. [Online]. Available: http://www.colostate.edu/ ypburns/monte/documents.html

Carlos Carreras received the Engineering degree in 1986 from the Universidad Politécnica de Madrid (U.P.M.) and the MS degree in 1989 from the University of Texas at Austin, both in Electrical Engineering. He obtained his Ph.D. in Electrical Engineering in 1993 from the U.P.M. He has worked as staff engineer for Bull-Madrid in 1987 and for Schlumberger-Austin from 1989 to 1991. Since then, he is an Associate Professor of Digital Electronics and Systems at the U.P.M., in Madrid, Spain.

Ian D. Walker received the B.Sc. (1983) in Mathematics from the University of Hull, England, in 1983, and the MS (1985) and Ph.D. (1989), both in Electrical Engineering, from the University of Texas at Austin. From 1989 to 1997 he was in the Department of Electrical/Computer Engineering at Rice University, in Houston, Texas, where he held the positions of Assistant and Associate Professor. Since 1997 he has been an Associate Professor in the Department of Electrical/Computer Engineering at Clemson University, in South Carolina.