MISC'99

PREPRINTS

Workshop on

Applications of Interval Analysis to Systems and Control

with special emphasis on recent advances in Modal Interval Analysis


Organised by:
Dep. d’Informàtica i Matemàtica Aplicada,
Institut d’Informàtica i Aplicacions,
Universitat de Girona.

Sponsored by:

MONET

Fundació Privada: Girona, Universitat i Futur

Patronat de l’Escola Politècnica Superior de la Universitat de Girona

Generalitat de Catalunya
ROBOT RELIABILITY ESTIMATION USING INTERVAL METHODS

Carlos Carreras*, Ian D. Walker*, Octavio Nieto+ & Joseph R. Cavallaro*

+Departamento de Ingeniería Electrónica. E.T.S.I.Telecomunicación
Universidad Politécnica de Madrid
28040 Madrid, SPAIN
carreras@die.upm.es, nieto@die.upm.es

*Department of Electrical and Computer Engineering
Clemson University
Clemson, SC 29634 USA
ianw@ces.clemson.edu

*Department of Electrical and Computer Engineering
Rice University
Houston, TX 77005 USA
cavallar@rice.edu

Abstract

Reliability is a critical issue for many robotic applications, where failures can be disastrous. Traditional reliability analysis algorithms depend on the algebraic manipulation of given input data (representing subsystem and component reliability). However, this information is particularly difficult to obtain in robotics, where the data is typically uncertain at best. Thus there is inherent uncertainty in robot reliability estimates. In this paper, we discuss the applicability and usefulness of interval techniques to the robot reliability problem. Through a canonical example, we show how the use of a novel interval method can significantly improve the accuracy of robot reliability estimates over existing methods. The approach is shown to be more efficient and effective than using corresponding integer methods.

Keywords: Robotics, interval methods, probabilistic estimation, reliability, environmental restoration.

1 Introduction

The last few years have seen an unprecedented growth in the spread of technology into new and unexplored arenas. As technology is used more widely, issues of reliability, safety, and fault tolerance are assuming ever greater significance.

In the field of robotics, a key driver for the development and application of new technology has been the need to send robots into hazardous and remote environments. These environments include the space, nuclear, and undersea arenas. A key difficulty in these areas is that a failure in the robot can have damaging or disastrous consequences, and repair is difficult and often impossible. Thus reliability is necessarily a major concern for these applications.

Largely motivated by the above concerns, there has been much research in the area of reliability and fault tolerance of robots in the last few years [15]. This work has concentrated on both fault detection and isolation, and on reliability modeling and analysis, the topic of this paper.
Estimates of the overall reliability of a robot are important for determining factors such as length of operation time, times between maintenance, and even whether deployment is feasible for certain applications. For reliability analyses, the primary tool used in robotics has been Fault Tree Analysis [15]. With strong motivation and funding coming from the nuclear and hazardous waste clean-up communities in both the USA [13] and Europe [9], numerous Fault Tree studies have been performed recently for different types of robots [9, 16].

Fault Trees are an established reliability analysis tool, and have been utilized successfully in the nuclear and aerospace industries, for example. However, quantitative reliability estimates using fault trees (or other established reliability methods) rely heavily on reasonably accurate input data (in this case, the mean time to failure or failure rate probabilities of subsystems and components). In the case of robotics, this data is at best often poorly known and thus the results of conventional Fault Tree Analyses are drawn into question [17]. Methods for coping with this uncertainty have been proposed [12, 14, 17]. However, these methods are either overly simplistic or far too complex to be of value in practice. A different way to look at the problem needs to be found.

Over the last few years, there has been increasing interest and activity in the theory and application of Interval Arithmetic and Interval Computation [10, 11]. These techniques arose from the need to deal with uncertainty (manipulate imprecise data, keep track of round-off error, etc.). Interval methods have been applied to numerous practical applications recently [1]. The ability of interval methods to keep track of ranges of data through computations in a rigorous manner is very appealing from the point of view of robotics, where there are numerous key problems involving manipulation of uncertain data via structured algorithms.

There have been several works recently proposing interval methods for various aspects of robotics. In [5], an interval approach to solving inverse kinematics (which finds the configuration of a robot given the end effector location in the workspace) was introduced and analyzed. The application of interval methods to the control of a mobile robot was reported in [19]. These works demonstrate that robotics is an area in which interval methods can be effective and useful. However, none of these works considered the important robot reliability problem, as discussed in this work.

In this paper, we apply a novel interval method to a key problem in robot reliability, that of estimating the reliability of a robot given (uncertain) data on the failure probabilities of its key subsystems. We show how, by the use of this method it is possible to significantly improve the resolution and estimate of the reliability compared to previous approaches, and also compute the estimate more efficiently than using integer-based approaches. This is an important result, and opens us the possibility of quantitative robot Fault Tree analyses in cases which had previously been thought inaccessible.

The paper is organized as follows. In the following section, the robot reliability problem is reviewed, and a specific representative example, with its attendant algorithm detailed. We then introduce a novel interval method, for which the robot reliability algorithm is seen to be ideally suited. Results of application of our interval method to the algorithm are given in section 4, with discussion and conclusions following.

2 The Robot Reliability Problem

Robots, particularly robot manipulators, are complex electromechanical systems. They are subject to numerous failure modes, from sensors, actuators, drive systems, etc., to human operator errors. These failures can directly cause, or interact together to cause, the failure of the robot system itself [15].

To analyze this complex interaction of faults, a systematic tool is needed. In the robotics arena, the tool which seems best suited (and has been almost universally adopted in robot reliability analyses) is that of Fault Tree Analysis [15]. Fault Trees provide, via a tree structure, a logical interconnection of events (faults) which can cause the failure of the ‘top event’ (base of tree). Basic events (the ‘leaves’ of the tree) are faults or failures in subsystems or components of the system. These basic events are combined through the branches of the tree using logic gates (typically AND and OR gates), according to the structure and operation of the system. The final result is a tree representing the logical flow of fault and failure events affecting the top event.
For example, consider a segment of a three joint robot arm operating in the plane (see Figure 1). This is a fairly typical type of robot 'forearm' being used for hazardous waste clean-up [18].

![Three joint planar robot with redundant sensors](image)

Figure 1: Three joint planar robot with redundant sensors

The arm has one actuator and dual redundant sensors (either sensor alone is sufficient for effective operation of the joint, e.g. dual position sensors, or a position and a velocity sensor) at each joint. The robot thus has redundancy in both sensors and joints, since it can tolerate one sensor failure at any joint, and also any single joint failure: it will cease to be able to position its distal link (the one furthest from the base) arbitrarily in the plane of its motion (it's primary task) if it suffers more than one joint failure. We assume that failed joints are braked, to allow the remaining ones to continue. Thus overall robot failure happens if appropriate combinations of sensor and/or motor failures occur.

Failure logic for the above description of this robot can be represented by the fault tree given in Figure 2 [17].

![Fault Tree for three joint planar robot with redundant sensors](image)

Figure 2: Fault Tree for three joint planar robot with redundant sensors

In the figure, $J_i$ indicates joint $i$, $M_i$ actuator $i$, and $S_iA$ and $S_iB$ the primary (A) and backup (B) sensors of joint $i$. Note the AND gates model the redundant safety systems (dual sensors, redundant joint) built into the robot. For a more detailed analysis, the 'leaf' nodes of this tree can be expanded further (to the level of detail desired by the analyst). However, this example is representative of the key issues and problems associated with robot fault tree analysis, and has been used previously [16] as a canonical example for new robotic fault tree analysis techniques.

The main feature of fault trees from a reliability viewpoint is that it is straightforward, using the structure of the trees, to compute the failure probability of the top event ($s$) given failure probabilities for the basic events. Given $s$, the reliability $R$ of the robot is

$$R = 1 - s$$
For the fault tree given in Figure 2, an algorithm to compute the probability of failure of the top event (s) from the inputs \((a, b, c, d, e, f, g, h, i)\) is given by (for more details on the procedure, see [17])

\[
\begin{align*}
  j &= a \times b \\
  k &= c \times d \\
  l &= e \times f \\
  m &= g + j \\
  n &= h + k \\
  o &= i + l \\
  p &= m \times n \\
  q &= m \times o \\
  r &= n \times o \\
  t &= p + q \\
  s &= r + t
\end{align*}
\]

where \(a\) and \(b\) are the (failure probabilities of the) primary and redundant sensors on joint 1, and \((c, d)\) and \((e, f)\) are the corresponding primary and redundant sensor failure probabilities for joints 2 and 3, respectively, and \((g, h, i)\) are the failure probabilities associated with the actuators of joints \((1, 2, 3)\), respectively. (Note: the above algorithm assumes the so-called "rare event approximation" \(P(AUB) = P(A) + P(B)\). This estimate, which is conservative and reasonable for most robot applications, is typically made in practice [17]).

Interesting subcases of the above algorithm occur when: (i) \(a = b = c = d = e = f, g = h = i\) (all sensors and all motors are grouped together, i.e., this corresponds to a scenario where a common event - such as an excessive radiation burst - causes the failure of all sensors or all motors - this produces essentially a two-input case for the algorithm); (ii) \(a = c = e, b = d = f, g = h = i\) (a three-input case, corresponding to the sets of primary and redundant sensors each having common failures to others in their own set, but the primary and backup sets having different failure characteristics); (iii) \(a = b = c = d = e = f, g, h, i\) independent (4 inputs); and (iv) \(a = c = e, b = d = f, g, h, i\) independent (5 inputs). Cases (i) and (ii) will be used later to obtain exact results for the output, in order to investigate the errors of interval and integer approaches for this algorithm. Cases (iii) and (iv) represent increased complexity in the type of faults considered, though still short of the full algorithm (with 9 inputs).

If sufficiently good statistical data for the failures of the basic events exists, the above procedure yields a very good estimate of overall system reliability. While this has been the case for other applications (numerous examples can be found in the nuclear and aerospace industries) for robotics applications subsystem failure probabilities are almost always poorly known [17]. This is due to the fact that robots have typically been 'one of a kind' or 'several of a kind' devices for which good statistics are not available. In addition, for robots operating in remote and hazardous environments, previously existing failure statistics (typically for more benign environments) for the robots or their components cannot be directly applied to the deployed robot. Thus any single number resulting from such a calculation as discussed above is typically not trusted in robotics.

Researchers have tried to address this issue by computing some type of top event 'output distribution or density', taking into account the uncertainty inherent in the input data for the calculation. For example, in [12] the use of input probability distributions was proposed, and an approach to formally computing the statistics of the output probability distribution proposed. However, such formal stochastic techniques require better input statistics than available in robotics, and are also typically computationally intensive.

A different approach was proposed in [17], in which the raw input data was expanded into ranges (essentially the input data was treated as 'fuzzy', and the input and output distributions became fuzzy 'possibility distributions'). Given an estimate of the mean value of the probability of failure of an input component, and basic information about the statistics for that component, the inputs were represented as trapezoids as shown in Figure 3.
Figure 3: Representation of input data as fuzzy possibility

In Figure 3, the possibility of the input failure probability is represented (note that the probability has a possibility of 1 over a range of values of p, from v to w). Between u and v, and between w and x, the possibility varies linearly between 0 and 1. Thus each distribution is uniquely described by the set \((u, v, w, x)\).

This approach is well-suited for representing the uncertainty in statistics of (inherently poorly known) processes. For example, existing reliability data for electric motors (typically used in robot manipulators) across a wide range of applications reports an overall failure rate (probability) \(\bar{p}\) of 0.00924, together with information that 68 per cent of cases will be between 0.22 and 4.5 times this value, and 90 per cent of cases will be between 0.08 and 11.8 of it [17]. Thus, for this case, \((u, v, w, x)\) are estimated as \((0.000739, 0.00203, 0.04158, 0.109032)\), so that 90 per cent of the cases are represented under the "possibility distribution".

To compute the reliability estimates for the robot in [17], the trapezoidal estimates were propagated through the fault tree algorithm using a simple approach, first proposed in [14], where the output distributions are computed (pairwise) from the corners of the input trapezoidal distributions:

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

where the input distributions are given by \((u_1, v_1, w_1, x_1)\) and \((u_2, v_2, w_2, x_2)\). This approach allows the incorporation of some additional uncertainty information in the output distribution, beyond the simple propagation of the means.

The approach in [17] and summarized above is computationally simple, and produces some new insight, but the resulting output distribution is known to be extremely conservative (as we shall see later). In this paper, the robot problem (planar three joint robot, electric motor actuators, and dual optical encoders at the joints) is taken to be the same as that in [17], but the input data is reformulated as an estimate of the probability densities of the inputs. For example, for the electric motor data discussed above, an input distribution (histogram) is constructed as follows:

\[
g = (0, 0.000739) : 0.05 \\
g = (0.000739, 0.00203) : 0.11 \\
g = (0.00203, 0.00924) : 0.34 \\
g = (0.00924, 0.0416) : 0.34 \\
g = (0.0416, 0.11) : 0.11 \\
g = (0.11, 1) : 0.05
\]

Here the interpretation is that the function \(g\) (representing the failure probability of the actuator of joint 1) has 0.05 of its probability mass between 0 and 0.000739, etc. As discussed in the following section, in our new approach we transform the above types of input distributions into a form suitable for the application of an interval approach. The interval approach is then used to estimate the output distribution of the algorithm (estimate of probability statistics of overall robot failure). This allows us to obtain significantly better estimates of the overall robot reliability than with the previous method.
To the best of our knowledge, no-one has previously proposed using interval analysis to address the problem. This is somewhat surprising, as the problem (characterized by uncertain inputs to a structured algorithm) is of a form that seems ideal for interval arithmetic. In the following, we discuss the application of a novel interval simulation method to the above robot reliability problem.

3 A Novel Interval Approach

The robot reliability problem can be viewed as a particular case of the general problem of evaluating the result of a sequence of mutually dependent arithmetic functions given the distributions (obtained here from probability density estimates) of the initial input variables. While in some other estimation problems computing an exact result by theoretical calculation might be complex but possible, the uncertainty of the input probability densities in the robot reliability problem discards this approach in practice. In fact, these densities can naturally be expressed in the form of histograms [17], thus supporting an interval approach. On the other hand, traditional estimation methods based on simulating random input samples can be hardly trusted if sample spaces are large and estimation times must be short.

Two major difficulties are encountered when using an interval approach to solve this problem. One arises from the fact that the arithmetic functions involved are mutually dependent [7]. Such dependency is the reason why the independent evaluation of the functions does not yield meaningful results as unfeasible values are propagated through the computation. The other problem is that, if no approximations are taken into account, the number of intervals that can be generated from the computation can grow up to a point that the computation using intervals has similar complexity than other traditional simulation methods based on discretized input values, thus eliminating the advantages of the interval approach.

For the situation discussed in this paper, where input distributions are expressed in the form of histograms, a novel interval method that takes dependencies into account and limits the number of intervals involved in the computation has been used for estimation [3, 2, 4]. This method is intended for integer computations and embeds concepts from the field of Abstract Interpretation techniques [6, 8]. These techniques, which are based on the definition of abstract domains and abstract operations, are frequently used in compiler optimization for fast analysis of algorithm properties. As shown below, results indicate that this method, not only provides better estimates than traditional approaches when the same computation time is considered, but it allows, through its particular interval model, that much larger input spaces can be exhaustively evaluated in short computation times, thus eliminating the errors due to limited random sampling.

Since the interval method is intended for integer computations and the input distributions of the reliability problem are defined for real numbers in the range [0, 1] a transformation of the latter is required before the actual computation is performed, as discussed below. This transformation implies the discretization and scaling of the distributions, together with the addition of a few division operations in the reliability algorithm to preserve its behavior after the scaling.

The discretization step is aimed to obtain a finite input space for the computation. The range [0, 1] is partitioned into a finite number of disjoint subranges of equal size. Then, each subrange becomes a possible input value that can be represented by a number (its midpoint), and a probability (the subrange probability in the original distribution.) The number of subranges (discretization steps) determines the accuracy of the estimation as well as the complexity of the computation. The scaling step transforms the range [0, 1] into the range [0, N – 1], where N is the number of discretization steps. The resulting real distribution can be directly mapped to an integer distribution in the same range (assuming that N is an integer), where each integer n has the probability of the subrange [n – 0.5, n + 0.5] in the real distribution. In the following paragraphs, it is assumed that the input distributions have been discretized and scaled accordingly, so the interval method can be applied on them.

The two key elements of the interval approach are the interval model and the computation algorithm. Both of them are based on the following definitions.

**Definition 1.** An interval [a, b]/p is the set of N = (b – a + 1) integers x that verify a ≤ x ≤ b with an associated probability of occurrence p which is uniformly distributed inside the interval, so that the probability of any x is p/N.
Definition 2. A maxinterval is an interval $[A, B]$ which verifies one of the following identities.

$$
[A, B] = \begin{cases} 
[0, 0] & N = 1 \\
[-2^{-n} + 1, -2^{-(n+1)}] & N = 2^{-n} (n < 0) \\
[2^{-(n-1)}, 2^n - 1] & N = 2^{n-1} (n > 0)
\end{cases}
$$

where $n$ is an integer and $N$ is the number of integers included in the maxinterval.

It can be observed that the set of maxintervals obtained when considering all possible values of $n$ constitutes a partition of the integer axis into disjoint sections with exponential increasing sizes (powers of 2) as they depart from 0.

In this situation, the interval model is defined by means of two constraints which are enforced in the internal representation of intervals throughout the computation: (1) the size of any interval is limited so that there is a maxinterval that includes it, and (2) any two intervals included in the same maxinterval representing values of the same variable must be merged into a single interval. These two constraints imply that (1) if an interval crosses or exceeds the bounds defined for maxintervals, it must be represented by a set of contiguous intervals, each of them included in a maxinterval, and (2) the merging of two intervals included in the same maxinterval results in the minimum single interval that includes both of them with an associated probability obtained from adding their probabilities.

The probability density estimates used in the computation are represented in terms of the interval model which provides the means for an approximate representation in terms of a reduced number of intervals. While this approximation can be a problem in some cases, it is mostly appropriate for the reliability problem where the interesting features of the probability distributions appear very close to the origin (typical input failure probabilities are quite small, of the order of $10^{-3}$ or less), the region where maxintervals are smaller thus allowing greater accuracy in the results. In general, the model provides large reductions in input spaces if large input ranges are the major contributors to the space size. However, the impact of the model is reduced if the size of the input space is dominated by a large number of input variables with small ranges.

Once the input distributions have been transformed to integer ranges and represented in terms of the interval model, the computation method must calculate the output distributions (estimate of probability density for overall failure of the robot) through repetitive execution (simulation) of the reliability algorithm with different combinations or vectors of input intervals. This is the approach used here to avoid the dependency problem previously mentioned.

Standard definitions of operations between intervals are used for the computation [3]. They provide single output ranges (including all possible integer results) which are computed from the endpoints of the interval operands. In particular, considering positive intervals:

$$
\begin{align*}
[x_1, y_1] + [x_2, y_2] &= [x_1 + x_2, y_1 + y_2] \\
[x_1, y_1] - [x_2, y_2] &= [x_1 - y_2, y_1 - x_2] \\
[x_1, y_1] \times [x_2, y_2] &= [x_1 \times x_2, y_1 \times y_2] \\
[x_1, y_1] / [x_2, y_2] &= [x_1/y_2, x_2/y_1]
\end{align*}
$$

Division by 0 is a special case than can occur through the computation due to approximations implied by the interval approach. Such division is avoided by transforming it into a division by 1. Output probabilities are uniformly distributed in the output range and depend on the input probabilities of each particular run of the algorithm. It is clear that such uniformity in the output distributions implies an approximation that should be considered in the analysis of the estimates.

The computation method defines three phases in each run of the reliability algorithm: (1) selecting the input intervals from the input distributions, (2) computing the algorithm, and (3) collecting the output ranges of the run in global output distributions. The probability of each run is determined by the probabilities of its input intervals. This computation method is closely coupled with the interval model in that such model, not only is used for the input representation, but it is also applied when output ranges are collected. This is the key to avoid a combinatorial explosion in the number of intervals through the
computation because, by means of its second constraint, the interval model puts limits on the number of intervals in each output distribution.

While it is clearly preferred to simulate all possible combinations of input interval vectors in order to achieve accurate results, this is not always possible. In those cases where the input space of intervals is still too large to be exhaustively simulated despite the reduction implied by the interval model, running a sample of the input space as in traditional methods is also possible. In this case, while integer simulation (where input vectors relate to individual integer values) is usually based on random input generation, interval simulation is based on a different approach. Inputs are generated following a pattern that favors that most input combinations with high probabilities are computed first. This implies a sorting of intervals according to probabilities (not feasible with integers), and causes a faster reduction of errors due to limited sampling.

4 Results

Results are divided into two major groups. The first group evaluates the accuracy and behavior of the interval approach when applied to the reliability algorithm. This is done using reduced input spaces which allow exact computation of the output distributions for comparison. The second group comprises the reliability estimates provided by the interval method for the larger input spaces of general interest.

It is observed that, in order to maintain the accuracy of the available data in the input distributions, at least $10^5$ discretization steps must be used (the input data used, taken from [17], is estimated to 5 decimal places). Since the algorithm under study has up to 9 input variables, the total number of possible combinations of input values can be computed as $10^5 \times 9$. Therefore, in order to obtain exact output distributions to evaluate the interval model, the input space must be reduced. This reduction can be achieved either by reducing the number of steps or the number of variables by assuming that some sensors or motors respond identically (as discussed earlier). Considering the need for reasonable times for the exact computation, four cases have been considered in the first group of results. Two of them assume only two input variables (all sensors equal and all motors equal) and are modeled with 201 and 2001 discretization steps respectively. The other two assume three input variables (two types of sensors, $a = c = e$ and $b = d = f$, and all motors equal) and are modeled with 201 and 301 steps.

First, exact output distributions have been computed for each case. Then, the evolution of the error
Figure 5: Error evolution for three inputs and 201 steps

provided by the interval approach has been measured for increasing computation times until complete simulation of the input space. Finally, the evolution of the error produced by limited integer simulation with random input generation is similarly measured at increasing times. In this case, random generation produces a diversity of results, so each point in the plot is the average of the measurements taken from four different runs. Error bars at each of these points describe the variation of results obtained from these four runs.

Figures 4, 5 and 6 contain the results of the computations. Error $e$ is represented as a percentage of the exact output distribution. In particular, it is obtained as

$$\text{e} (%) = \left( \frac{\text{Exact} - \text{Measured}}{2 \times \text{Exact}} \right) \times 100$$

(6)

where \text{Exact} and \text{Measured} are the exact and measured output distributions in terms of the interval model, and the factor 2 accounts for the fact that each misplaced result causes a difference in the distributions of twice its probability. Time $t$ corresponds to CPU time and is measured in milliseconds on a logarithmic scale. Points in the integer simulation curves include bars which describe the variation of measurements obtained in the four runs used for the plot. This shows the uncertainty on the accuracy of this approach for short computation times.

Results considering two inputs (Figure 4) clearly demonstrate the advantages of the interval approach. They provide better estimates than integer simulation with random inputs for fixed computation times in all cases. In addition, they maintain a similar evolution of the error when the input ranges are increased one order of magnitude, while the error from integer simulations is significantly displaced towards larger computation times.

The results from simulations with three inputs are not so clear because less steps can be considered if computation times are to be maintained. Such smaller ranges do not provide the setup where the interval method has more advantages: large ranges. In fact, this is why no more than three variables have been considered to evaluate the interval method: more variables require many fewer steps for reasonable computation times up to a point that integer input distributions only have a few possible integer values. This makes the comparison of approaches meaningless since only a few intervals around 0, therefore very similar to individual integers, are used in the computation.

Figure 5 corresponds to three inputs and 201 steps. It shows that the limited input space for the interval computations can produce errors of the same order than integer simulation. However, the small
increase to 301 steps (Figure 6) already shows an improvement in the evolution of the interval curve.

while some displacement towards higher computation times is detected in the integer curve. Therefore,
it can be inferred that the reliability estimation based on, at least, $10^5$ steps will clearly show an advantage
for the interval method. This is confirmed by the second group of results explained below.

Table 1 collects some interesting data from the previous experiments. It includes the sizes of the
input spaces for the integer and interval exhaustive computations in terms of the numbers of input integer
vectors and input interval vectors respectively, the corresponding computation times in milliseconds,
and the final error obtained from the interval approach. As previously described, this error is due to
the approximations included in the method. The numbers show the great reductions allowed by interval
simulation, while final errors maintain acceptable values for fast estimation. Although these results seem
to indicate that errors increase with the number of steps, results from additional experiments do not
support this theory. For example, the error for two inputs with 1001 steps is 6.24%, and for three inputs
with 101 steps is 4.05%.

To explore the ability of the interval algorithm to estimate the output distribution (second group of
results), several tests were conducted. The cases of 4, 5, and 9 inputs (the full case) were investigated for
different numbers of inputs and steps. In particular, $(10^5 + 1)$ and $(10^6 + 1)$ steps were used to transfer
the input data to integers. Note that this level of complexity of inputs, which is prohibitive for effective
integer simulation, is required in order to preserve the information in the failure probability input data.
Actually, feasible computation times only allow for integer simulations with percentage coverages below
$10^{-10}$ of the input space in all cases, so integer results can hardly be meaningful.
Figure 7: Results for 9 inputs after $10^4$ CPU seconds of computation time

Figure 8: Complete results for 5 inputs (Computation time = 3955 sec)
The results of the interval estimates for the overall robot failure \( o \) for the above example are given in Figures 7, 9, 8, and 10. These figures concentrate on the results around 0 which is the area of interest for the problem. For larger values of \( s \), the distributions gradually converge to null probabilities. Figure 7 displays the results from the interval method after \( 10^4 \) seconds of CPU simulation time (0.0016\% of the interval input space). All the cases simulating five and four inputs (as discussed earlier) allow for complete simulation of the input space and the results for \( 10^5 \) steps are shown in Figures 8 and 9. Notice that for all these cases we have \( (g, h, i) \) independent, as compared with the cases of two and three inputs (where \( g = h = i \)) used for the exact estimates. This is reflected in the resulting distributions in the way that the results for more inputs displace away from 0, which is to be expected since there are a greater number of paths to failure. The plot for the case with \( 10^6 \) discretization steps (Figure 10) highlights structure around zero, and displays a definite maximum away from zero.

For the same example (with 9 inputs), the output distribution trapezoid was calculated in [17] as

\[
(u, v, w, x) = (0.000000164, 0.0000125, 0.00648, 0.062)
\]

This represents the previous 'state of the art' of robot reliability output distribution estimation [17]. Note that the distribution is a simple trapezoid, which is known to inherently overestimate the reliability (note that this analysis estimates what is 'possible' using fuzzy logic, rather than what is 'probable' via probabilities). However, analysis of the extent of this overestimate is difficult, and has not been conducted for the robotic case. Therefore, it was not previously known how conservative the 'fuzzy possibility' approach in [17] was.

From the estimates using the interval approach, several important points can be made. First, the interval methods correctly pick out the expected structure of the output distribution (with most of the probability 'mass' near zero, a 'peak' away from zero, and a strong decay towards 1). In addition, it is seen how the 'fuzzy possibility distribution' above from [17] grossly overestimates the actual probability distribution, as represented by the interval estimates. Notice that, for a scale of \( 10^5 \), the trapezoidal distribution above estimates the 'possibility of probability' of robot failure to be one for values of \( o \) as high as 0.00648. Comparing this with the distribution estimates in Figure 7, we see that the majority of the probability mass is well below this point.

This is very important practically, as the 'drop-off' behavior of the estimate is the critical indicator that will determine whether the system is considered reliable enough for deployment. The estimates from the interval method give a better insight into the true character of the output distribution, which will lead in turn to better estimates of the overall reliability. Notice that the interval method implicitly represents a refinement of 'possibility distribution' in [17], since the 'possibility distribution' is based on a small sampling (4 points, corresponding to the first two 'confidence intervals' of the input data) of the calculations used to find the interval estimate.

To the best of our knowledge, the results in this paper are the first attempt to estimate a failure density for a robot manipulator from a fault tree model. Previous efforts have only considered a single integer calculation (effectively finding the 'mean'), or projected simple ranges as in [17]. The ability of output probability density estimates to better reveal the underlying structure than fuzzy or moment-based approaches provides a strong motivation for the estimation of probability densities in general. However, due to the complexity of the inputs inherent in the robot reliability problem, integer simulation is not a viable option in this case (representation of the inputs and computation times become prohibitive). Thus, the interval approach seems the most natural way to estimate these types of reliability distributions for the robotic case.

Therefore our results not only indicate the relevance of interval methods to robot reliability, they also in this instance immediately shed new light on the structure of the solution itself. We anticipate that the use of interval methods in this way can transform the nature of robot reliability analyses. In continuing work, we are applying the interval methods to more complex robotic examples and fault tree algorithms. We are also considering the applicability of the method to algorithms arising from alternative robot reliability techniques.
Figure 9: Complete results for 4 inputs and \((10^5 + 1)\) steps (Time = 219 sec)

Figure 10: Complete results for 4 inputs and \((10^6 + 1)\) steps (Time = 424 sec)
5 Conclusions

In this paper, we have considered the application of interval methods to problems in robot reliability. Robot reliability algorithms appear well suited to interval methods due to their inherent structure and uncertain inputs. We have considered, for a specific example, the application of a particular interval method to estimate the output distribution (overall robot failure probability density) of a robot reliability method (quantitative fault tree analysis). The interval method used has the features of an interval model, which constrains the number and the sizes of the intervals used to represent probability distributions in the form of histograms, and an algorithm that allows the computation of output estimates. The results clearly demonstrate the power and advantages of the interval approach, producing results which are less conservative and which provide more structural insight than with previous approaches.

Acknowledgments

The first and third authors are supported by projects ESPRIT 24137 and HCM CHRX-CT94-0459. The first author is also supported by project CICYT TIC97-0928. The second author is supported by the National Science Foundation under grant CMS-9796328, and by the U.S. Department of Energy (through Foster-Miller Inc.) under contract #DE-FG07-97ER14830, subcontract #805-05002.

References


