RICE UNIVERSITY

Robot Reliability Through Fuzzy Markov Models

by

Martin Leuschen

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Master of Science

Approved, Thesis Committee:

Dr. Joseph R. Cavallaro, Co-Chairman
Associate Professor,
Electrical and Computer Engineering

Dr. Ian D. Walker, Co-Chairman
Associate Professor,
Electrical and Computer Engineering

Dr. Fathi Ghorbel
Assistant Professor
Mechanical Engineering

Houston, Texas

April, 1997

Robot Reliability Through Fuzzy Markov Models

Martin Leuschen

Abstract

In the past few years, new applications of robots have increased the importance of robotic reliability and fault tolerance. Standard approaches of reliability engineering rely on the probability model, which is often inappropriate for this task due to a lack of sufficient probabilistic information during the design phase. Fuzzy logic offers an alternative to the probability paradigm, possibility, that is much more appropriate to reliability in the robotic context.

This thesis deals with the construction and interpretation of the fault tree and Markov model reliability tools in a possibilistic (fuzzy) context for robotics. Although fuzzy fault trees are well established reliability tools, fuzzy Markov models have not been used in this context. Additionally, the thesis shows how the possibilistic Markov model used in other contexts is inappropriate in the context of fault tolerance, as it does not preserve the uncertainty information contained in the input. A new reliability method involving the joint use of fault trees and Markov models under fuzziness is developed and applied to examples.

Acknowledgments

I would like to thank :

Dr. Cavallaro and Dr. Walker, for motivating this work and for their help and guidance,

Dr. Ghorbel for serving on the committee,

This work was supported in part by the National Science Foundation under grant IRI-9526363 and by DOE Sandia National Laboratory Contract #AL3017.

To Sarah,

For your support under stress.

Contents

Illustrations

Tables

Chapter 1

Introduction

1.1 Motivation

Robot reliability has increased in importance alongside several important robotic applications. Robots are being used for dealing with hazardous environments, such as radioactive areas, where human maintenance is not feasible. They are also being applied in situations where extreme delicacy and precision are required, such as medical applications. Faults in these applications are likely to be catastrophic. Additionally, as robotic technology improves, robots are moving off the assembly line and into complex, unstructured environments. In these unpredictable workspaces, faults are both more common and harder to anticipate. As an overall result of these advances in robotics, reliability and fault tolerance are becoming ever more important in the field.

The increasing desire to produce more reliable robots has created interest in several tools used in fault-tolerant design. Such reliability tools seek to evaluate the effectiveness of new designs. The extra components needed for fault-tolerant robot designs obviously add extra costs and extra possibilities of failure. Reliability analysis tools such as fault trees and Markov models are needed to give hard numbers showing that the benefits of the fault tolerant design are tangible and worth the effort. Unfortunately, the component failure rates used in these calculations are often very dependent on configuration and environment, and thus known only approximately during the design phase [2, 38]. Any single value applied to the failure rates is likely to give a result that is misleading. Some way of considering the full range of failure rates is needed to give a good idea of what is and isn't known.

The standard approaches of reliability engineering rely on the probability model, which is often inappropriate for this task [2, 4, 26, 38]. Probability mathematics is often complex and nonintuitive, resulting in difficult and poorly understood analyses. Additionally, probability based analyses usually require more information about the system than is known, such as mean failure rates, or failure rate distributions [2]. Commonly, this results in dubious assumptions about the original data. Thus, any single value or distribution applied to the failure characteristics is likely to give a result that is misleading. The probability paradigm is also inappropriate for small samples, making probability based analyses questionable for regularly customized systems such as robots [4]. Some way of considering the full range of possible failure rates is needed to give a good idea of what is and isn't known.

Fuzzy logic offers an alternative to the probability paradigm, *possibility*, that is much more appropriate to reliability in the robotic context [2, 4, 26, 38]. Possibility mathematics allows for a quantitative reliability calculations that preserve the uncertainty present in the original data. The possibility model deals with uncertainty in a way that avoids making unwarranted assumptions, and makes the consequences of the required assumptions clear throughout the analysis. Small samples are also easily handled by possibility [4], making it uniquely suited to prototyping and customized systems, which are important aspects of both reliability and robotics.

Fuzzy logic also improves reliability analysis through the concept of *utility*. Standard reliability models usually assume a binary representation of failure. A system is working or it is failed. A more flexible and realistic model allows systems to degrade slowly as well as failing instantaneously. Utility allows for easy representation of partial system failures [2]. Thus, there is considerable motivation to adapt the traditional probability based reliability methods to the fuzzy logic context.

Of the common reliability tools, only fault trees and their variants have been *fuzzified* to any great extent. However, while fault trees are very useful, they are somewhat limited in their applications. Partial failures, coverage, repairable systems, and other important reliability issues are not covered well by the fault tree approach [3, 13], although recent developments in fault tree analysis are expanding the range of application somewhat [11]. Markov modeling is a valuable tool for dealing with the above situations. However, fuzzy Markov models have not been explored in the reliability literature. Fuzzy Markov models outside of the reliability area have concentrated on a method involving the fuzzy integral, which will be shown to be inappropriate for reliability work.

1.2 Contributions of Thesis

The main contribution of this thesis is the development of the fuzzy Markov model as a effective and useful reliability tool in robotics. Various methods or fuzzification are evaluated, and the best method is expanded into a fuzzy reliability tool. The benefits of the Fuzzy Markov model are explained in detail, as well as the proper interpretation of the results. This tool is examined in regards to several examples and its ideal areas of application determined.

This thesis also investigates the combination of the Fuzzy Markov model and fuzzy fault tree techniques. These two techniques are shown to be more useful together than when applied separately.

1.3 Overview of Thesis

Chapter 2 describes the tools used in this thesis. Fuzzy logic, classical and fuzzy fault trees, and classical Markov models are all examined in detail. The fuzzy integral, a central component in previous fuzzy Markov models, is presented. Chapter 3 examines the fuzzy Markov model. The desired output and properties of the model are followed by a look at several possible methods of dealing with the fuzzy component. These are evaluated and the best one is chosen to be used on the examples. Chapter 4 shows the fuzzy fault trees and Markov models developed in previous chapters in the context of several classical example robots. The strengths and weaknesses of both the robots and the methods are examined. Chapter 5 applies these reliability methods to a real robot, the Modified Light Duty Utility Arm (MLDUA). Chapter 6 considers future research in the area of fuzzy Markov modeling, and the consequences of the work already done. Appendices of figures used in the reliability analyses and outputs thereof follow.

Chapter 2

Background and Related Work

2.1 Fuzzy Logic and Possibility

The fuzzy set theory introduced by Zadeh [41] is an intuitive and powerful mathematical tool. It is an extension of standard set theory in which partial membership in a set is possible. A standard, or *crisp*, set A defined over X can be represented by a characteristic function $\mu(x)$ such that $\mu(x) = 1$ if x$\in A$ and $\mu(x) = 0$ if x$\notin A$. In a fuzzy set, the characteristic function is replaced with a *membership function* that can have any value between zero and one. This number represents our 'belief' or 'the degree of truth' of the statement x$\in A$. The obvious advantage of this is that we are no longer limited to only true and false, but have access to all the shades of grey in between. There is tremendous representational power in this simple idea,

An example would be the set of things that are hot. A crisp representation of 'hot' might have $\mu(x) = 1$ iff x >40C. A fuzzy set for hot might be

$$\mu(x) = \begin{cases} 0 & : x < 30\text{C} \\ (x-30)/40 & : x \in [30\text{C},70\text{C}] \\ 1 & : x > 70\text{C} \end{cases}$$

Note how the fuzzy set offers a more flexible and accurate description of the linguistic adjective 'hot' than is possible with a crisp set. It takes into account that some temperatures are definitely or definitely not hot, and that there is a 'fuzzy' region in between these where the temperature can be described as hot with varying degrees of truth. Fuzzy sets have found many applications interpreting between human language and numerical values for this reason.

The standard set operations that are used on crisp sets are also used with fuzzy sets. The intersection operation is usually represented by the *minimum* of the two membership functions. It is obvious that this reduces to the crisp intersection if the sets are both crisp. Similarly, the union operator is commonly the *maximum* of the two functions, and the complement operation is often one minus the membership function. (Other definitions of these operations are possible, but these are the most common[2].)

summary:

Crisp Operation	Equivalent Fuzzy Operation
Union	$max(\mu_a(x), \mu_b(x))$
Intersection	$min(\mu_a(x), \mu_b(x))$
Complement	$1 - \mu_a(x)$

A 'fuzzy number' is a useful way to represent uncertainty or noise in data. A common way to define a fuzzy number is a triangular membership function with a peak at the titular value and a base width and position appropriate to the uncertainty involved. For large amounts of uncertainty, trapezoidal membership functions are often appropriate. This thesis uses trapezoidal membership functions to represent the failure rates of sensors and motors in a robot. The failure rates of such components can only be guessed very roughly before a model of the complete system is built for testing.

Mathematical operations on fuzzy numbers can be defined through the use of the *extension principle*, which can be defined as follows[2, 22, 34]:

Let u and v be membership functions, and $f(u,v)$ be a function mapping u and v to membership function μ. Then the fuzzy result $\mu(y) = f(u,v) = sup(min(u(a),v(b)))$ over a,b s.t. $f(a,b) = y$.

The extension principle makes intuitive sense. For all the sets of values of the fuzzy number that could combine under function f to give result y, one chooses the set with the largest (*sup* operation) membership in all the required values (*min* operation).

The extension principle provides a good basis for math over fuzzy numbers, but can be difficult to apply to the functional definitions of the membership functions. In addition, application of the extension principle usually results in nonlinear membership functions that are difficult to deal with. These problems can be avoided through the use of α-cuts[2]. An α-cut is defined to be the crisp interval where the membership function of a fuzzy set is greater than or equal to a crisp number, α, on the interval [0,1]. It is usually much easier to apply the extension principle to an α-cut than to the membership function itself. The α-cuts are crisp intervals where the *min* operation is bounded below by α, allowing us to find the extreme values of the new α-cut using standard interval arithmetic. Approximate membership functions can be constructed from a sampling of membership functions over the range [0,1] by α-cuts. Note that fuzzy numbers are already approximate enough that it is often appropriate to take only the zero and one α-cuts and make a linear approximation. This is called the 'triangular' or 'trapezoidal' approximation. It is usually a conservative approximation in that it overestimates the failure possibilities [34].

Let $[a_l, a_h]$ and $[b_l, b_h]$ be equivalent α-cuts for fuzzy sets A and B. Table 2.1 lists the results of several different fuzzy arithmetic operations formed on these cuts [2].

Note that it is assumed that all variables are on the interval [0,1], as this is the usual case in reliability. Without this assumption, some of these formulas are slightly more complicated.

A fuzzy set that will be used extensively in this thesis is *utility*. *Utility* is a fuzzy set defined over the possible states of the system, with membership in each state dependent on how useful the system is in that state. This concept is used in degradable systems such as the kinematically redundant manipulator where there are states that are neither fully failed nor fully working. In this situation, a failed joint on such a manipulator will often reduce the workspace significantly, but not fatally, resulting in a less useful, but still functional, robot.

2.2 Possibility vs. Probability in Reliability

Human language is not the only source of imprecision engineers have to deal with. Complex, unpredictable, hazardous, and untested systems and environments, are all common problems faced by robotics design teams. The value of a particular variable (such as a component failure rate) may not be known to within even an

Operation	Equation to find α-cut
$A + B$	$[a_l + b_l, a_h + b_h]$
$A - B$	$[a_l - b_h, a_h - b_l]$
$A \times B$	$[a_l \times b_l, a_h \times b_h]$
$A \div B$	$[a_l \div b_h, a_h \div b_l]$
A^B	$[a_l^{b_h}, a_h^{b_l}]$
$\exp(A)$	$[\exp(a_l), \exp(a_h)]$
$\exp(-A)$	$[\exp(-a_h), \exp(-a_l)]$

Table 2.1 Useful Fuzzy α-cut Operations on the [0,1] Interval.

order of magnitude [38]! Probability-based reliability techniques would assume a mean, a distribution, and possibly a random process to represent this uncertainty. Unfortunately, the uncertainty is in the *probability distribution* representing the failure variable. Probability is being used to estimate another probability. These estimates can easily be off by orders of magnitude from the real probabilistic characteristics of the system, and the characteristics of probability math make it difficult to represent this uncertainty clearly.

These incorrect assumptions can lead to a very misleading model of the system [2]. For example, consider a system design where there are two independent modes of failure. Due to the complexities of the system, hostile work environment, and lack of a prototype, the failure rates of these modes are only known to the nearest order of magnitude. (This is probably optimistic.) Analysis using the means of the expected failure rates implies that the two failures are equally likely, while in fact it is possible that one type of failure will be *one hundred times* more likely than the other. A more complex analysis that maintained the uncertainty of the system might note this possibility, but would state that it was very unlikely. This is not necessarily true - it is an artifact of the assumption that the mean failure rate is in the middle of the interval. This problem is prevalent enough in the reliability industry that it has its own acronym, CBIA, for Correct But Irrelevant Arithmetic [2].

Fuzzy sets are useful for dealing with any situation where the exact value of a variable is unknown. Instead of a guess of the value of the variable, or of a probability distribution of its values (which is also usually unknown, and often theoretically dubious), fuzzy logic deals with the *possibility* of the variable taking on a set of values. Possibility is a less constrained method than probability. Instead of trying to describe the random behavior of the unknown element, it tries to describe the relative 'truth' of the statement 'x might take on value y'. This is much more appropriate than probability in situations where there is little data known about the variable, as no extra assumptions have to be made [2]. Possibility is similarly useful for small samples, whereas probability math requires many samples to be valid [4].

For example, consider the situation above. The given data is that a certain failure rate is within a certain order of magnitude. The probability analysis of the failure rate had to make assumptions, such as the mean failure fate was in the center of the interval, in order to proceed. These assumptions are not justified, but they are necessary for any analysis to be made in this context. Possibility math does not need to make any assumptions. It can take the data given and propagate the uncertainty through the model without adding to it or hiding it, resulting in a final analysis that accurately reflects the initial uncertainty.

Another argument in favor of possibility deals with extremes of uncertainty. If we consider the continuum of what can be known about a value on the real line, we have the extremes 'nothing is known about the value' and 'the value is known'. Probability cannot deal with either concept easily, requiring Dirac δ - functionals to deal with the 'is known' and failing entirely to represent the 'completely unknown' case. Possibility can represent either concept with ease. This implies that possibility is superior to probability near these extremes, which many reliability applications frequently skirt.

In addition, possibility math is valid for small samples. If a single robot is to be made, it is not meaningful to talk about the distribution of robot failure rates -

there is only one rate! Probability models applied to deal with uncertainty in this system are misleading and theoretically suspect, as the law of large numbers does not apply. Possibility math doesn't need large numbers to be valid, and applies equally well to large and small samples. A strong argument for fuzzy reliability analysis in this context can be found in [26], where it is demonstrated that a fuzzy reliability analysis of the Chernobyl reactor would have revealed the unreliability that caused the recent disaster there.

2.3 Fault Trees

Fault trees are a common tool in reliability analysis. Basic events are connected through a series of logic gates to a terminal event that usually represents the failure of the system. The classic *And* and *Or* gates are the basic gates needed to represent most systems. Additionally, the N/M (N-out-of-M) gate is useful in the redundant systems to be considered in this paper. An *And* gate represents a so-called parallel system. All of the components must fail for the system to fail. An *Or* gate corresponds to a series system. The system fails if any of the components fail. An N/M system is a type of redundant system. N out of the M elements in the system must fail before the system itself fails.

If the probability of failure of all the parts on the 'leaves' of the tree are known, these probabilities can be propagated through the tree using the following rules:[2, 34]

- Or gate: $P_c = 1 - (1 - P_a)(1 - P_b)$

- And gate: $P_c = P_a P_b$

N/M gates are best decomposed into an equivalent set of *And* and *Or* gates. One *Or* gate is used, with its inputs being the $\binom{n}{m}$ possible N member combinations of the M inputs. For example, the failure probability for a *2/3* gate could be calculated as:

$$P_d = 1 - (1 - P_a P_b)(1 - P_a P_c)(1 - P_b P_c)$$

2.4 Fuzzy Fault Trees

The probabilities for the basic events in a fault tree are often not known with great accuracy. Fuzzy numbers are a natural way to represent uncertainties such as these. The failure values used by Walker et. al. [38] are a good example of this. The fuzzy representation of a failure probability can be propagated through a fault tree using fuzzy arithmetic as described above. The resulting fuzzy number will cover a range of possible results, giving an accurate view of what is actually known about the system. In other words, a *possibility distribution of probabilities* is used. Note that if the α-cut method is used, this is equivalent to doing a best and worst case scenario using the lowest and highest values on the interval, respectively. Note also that if only the above gates are used on membership functions that are nonzero only for valid probabilities ($0 \leq P \leq 1$), all the resulting membership functions will also only be nonzero over valid probabilities. We will refer to this property as the $[0, 1]$ interval property of fuzzy fault trees.

2.5 Limitations of Fault Tree Analysis

Fault trees are somewhat limited in their modeling ability. In general, the set-up of the fault tree is geared towards the terminal event only. A system with several interesting and distinct states requires a fault tree for each state. This makes fault trees less useful in analyzing multi-state systems. Fault trees also suffer when dealing with redundancy that is neither exactly parallel or series. The 2/3 gate, for example, is a misleading model of the kinematically redundant robot, as it doesn't account for the decreased utility of the robot when only one joint is damaged. Although advanced fault tree techniques exist or are being developed to deal with these problems [11], these types of problems are more naturally dealt with using the technique of Markov modeling.

2.6 Markov Modeling

Markov Models treat a system as a series of states with specific, constant rate transitions between them. At all times, the system is in exactly one state. (Transitions are considered to be instantaneous.) The only information available is the current state, the allowed transitions, and the probability of these transitions. Such a system is referred to as memoryless, and is said to possess the *Markov property.* This means that the system is totally characterized by its current state. None of the past states or transitions have any effect on the transitions out of the current state.

A useful way of looking at Markov models is to consider a large population of such systems. The probability of being in each state will be roughly equivalent to the relative numbers of systems in each state in a large population. Thus 'the probability of being in state X at time T' is interchangeable with 'the population of state X at time T'.

A simple Markov model for a repairable one component system is shown in figure 2.6.

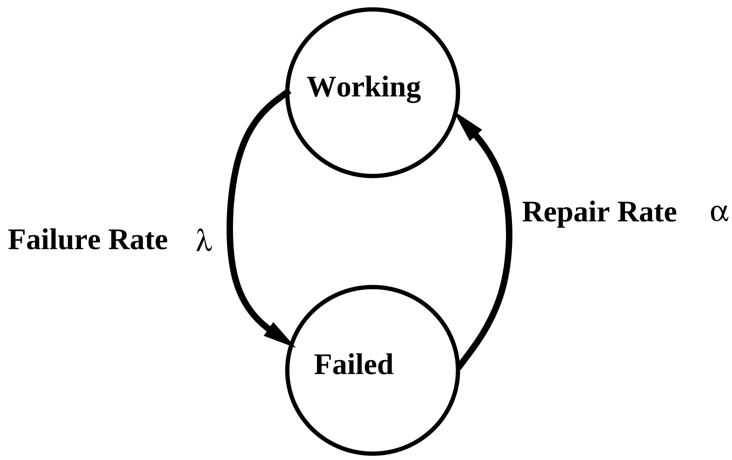

Figure 2.1 Markov Model for a Repairable One Component System.

The system fails with constant rate λ while it is in the working state. Once failed, repairs proceed at rate α. This system exponentially approaches the steady state where it has probability $\alpha/(\lambda+\alpha)$ of being in the working state and probability $\lambda/(\lambda+\alpha)$ of being in the failed state.

If the system is started with different populations than the steady state, the transient can be used to gain valuable information. If the state starts with population one in the working state, one can observe the expected behavior of a new system. One that begins in the failed state can give information on repair times and expectations. For configurations where the system is non-repairable ($\alpha = 0$), the transient is the only important result, as it shows the failure characteristics of the system.

The solution to this system is relatively easy to derive using differential equations. For discrete systems, such as the digital controllers of robots, difference equations or discrete simulations become more appropriate. These approaches are virtually identical for small time steps, so the simplest one to implement, discrete simulations, was used in the Markov modles in this thesis.

2.7 Fuzzy Markov Modeling and Fuzzy Integrals

As previously mentioned, component failure rates can be very difficult to calculate accurately during the design process, as environmental factors and component interactions cannot be easily determined before several prototypes are built [38]. This can lead to crisp values being given for an order of magnitude (or worse!) estimate if probability-based methods are used [2]. This problem is big enough to overwhelm the benefit derived from using Markov models to better represent the system.

However, the solution to fuzzy Markov models is not as straightforward as the solution to the fuzzy fault tree. The fuzzy fault tree dealt with arithmetic operations between different fuzzy possibilities only. The extension principle was sufficient for handling these operations appropriately, and gave useful results. (More on this in chapter 3.) Fuzzy Markov models require the evaluation of the possibility of a fuzzy event. This adds an extra element of fuzzyness to the system which must be dealt with.

All previous work known to the author regarding fuzzy Markov models has used the fuzzy integral to deal with this problem. A *fuzzy integral* is a mathematical operation between a fuzzy set and a fuzzy measure yielding a crisp result. They are useful to us because they can be used to determine a possibility of a fuzzy event, similar to the integration of a crisp event (interval) against a probability density function.

To use fuzzy integrals, one needs the concept of a *fuzzy measure* [8, 42]. Let $X = \{x_1, x_2, ..., x_n\}$. A fuzzy measure g on X is a set function mapping on the power set of X, $\mathcal{P}(\mathcal{X})$ (X and all possible subsets of X) to the interval $[0, 1]$:

$g : \mathcal{P}(\mathcal{X}) \rightarrow$ [0,1]

A fuzzy measure has the following additional properties:

(i) $g(\emptyset) = 0$.

(ii) $g(X) = 1$.

(iii) $\forall A, B \subseteq X$, if $A \subseteq B$ then $g(A) \leq g(B)$.

A fuzzy measure can be seen as a possibility analogue of standard probability measure, with the traditional additivity requirement replaced by the (weaker) condition of monotonicity [8]. Although there is nothing explicitly fuzzy about the definition, this concept is needed for fuzzy integrals and actually works very well in the possibility context.

The Sugeno fuzzy integral of fuzzy set $h(x)$ on fuzzy measure g is defined in the following way [8]:

$S_g(h) = \int h \circ g = \sup_{0 \leq \alpha \leq 1}(\alpha, g(H_\alpha))$,

where $H_\alpha = \{x \in X | h(x) \geq \alpha\}$, i.e. H_α is the α -cut of h.

The other commonly used fuzzy integral is the Choquet integral [8, 23, 42]. Using the notation developed for the Sugeno integral, the Choquet integral is:

$E_g(h) = \int g(H_\alpha)d\alpha,$

Either integral can be used for fuzzy possibility. The Sugeno integral is somewhat easier and more established in the community, while the Choquet integral has certain theoretical advantages [8, 23].

The usefulness of the fuzzy integral in possibility can be seen clearly in the following example. Consider some arbitrary possibility distribution f, defined over X. Although this distribution gives us a possibility for every element x_i, it does not explicitly give a possibility for subsets of X. However, if we wished to define such a function in a reasonable way, we would likely consider the definition of g to be a reasonable set of requirements. A common way of realizing this is to set $f(A) = \max_{a \in A}(f(a))$.

However, for fuzzy Markov models, we have to deal with *fuzzy* subsets of X. A standard fuzzy approach would be to take the maximum of the intersection of the fuzzy event set and the fuzzy possibility set, $\max_{x \in X}(h(x) \wedge f(x))$ (where $h(x)$ is the fuzzy event), as follows:

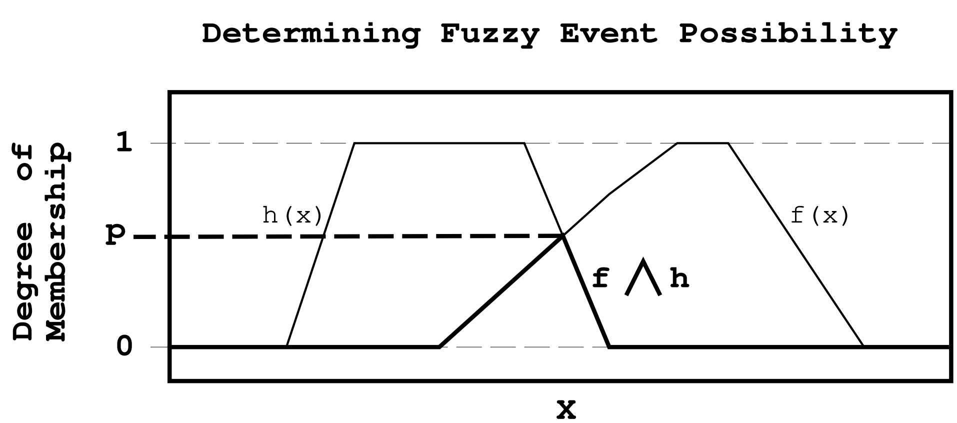

Figure 2.2 Finding the Possibility p of Fuzzy Event h Under Possibility Distribution f.

This is exactly the (Sugeno) fuzzy integral under our definition of g, however! The fuzzy integral is therefore a way of determining the fuzzy possibility of a fuzzy event. For our purposes, the possibility is the transition rate between two states of a Markov model, and the fuzzy event is the population of the original state. Thus, using this method, fuzzy Markov models are relatively easy to evaluate. Unfortunatly, as will be seen in the next chapter, this particular fuzzy Markov model is not the one we want to develop for reliability applications.

Chapter 3

Construction of a Fuzzy Markov Model for Reliability

3.1 The Format of the Model

A classical reliability Markov model breaks the possible configurations of the system into a number of states. Each of these states is connected to all the other states by a crisp *transition rate*. (This can be zero to represent impossible transitions.) The probability of being in each state (or *population* of that state) evolves over time according to these rates. An example Markov model output for a non-repairable system can be seen in figure 3.1.

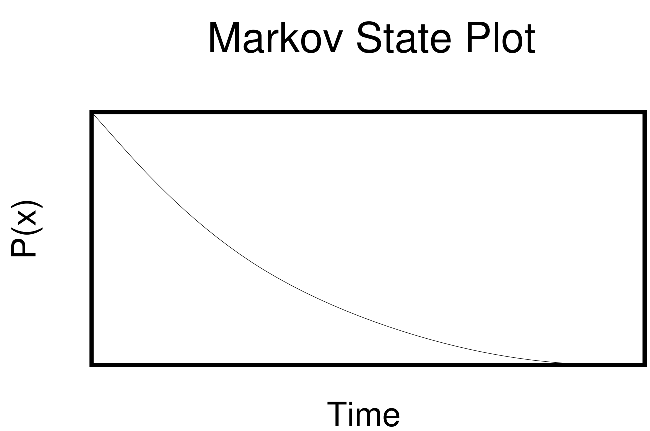

Figure 3.1 Markov Model Output for a Single State.

For our reliability *fuzzy* Markov models, both the populations and the transition rates will be fuzzy. These represent the possibility that the given probabilities are correct. Although this layering of possibility over probability may seem unorthodox, it actually makes a lot of sense.

The Markov model is a way of determining system behavior by using information about certain probabilities of events within the system. However, in reliability, we often don't know these probabilities, so we have to make some sort of guess. A standard approach is to guess a single crisp probability and assume it's good enough. As discussed in Chapter 2, this assumption is often wrong. A more sophisticated approach would be to assign a probability distribution to each of these probabilities, resulting in probabilities of probabilities. In addition to being convoluted, this approach assumes knowledge of the distributions in question, which is usually a bad assumption. This is also discussed in Chapter 2.

Our approach is to estimate the conservative and optimistic bounds of the probabilities in question, and use them to define ou trapezoidal membership function. This estimate is reasonably easy to perform for most systems, and has the benefit of being clear cut and easy to understand and modify. For simplicity of both mathematics and output, we use the trapezoidal membership function for our possibilities, using the conservative bounds for the zero-cut, and the optimistic bounds for the one-cut. The resulting output for our fuzzy Markov model is three dimensional, with axes of probability, possibility, and time. However, this can be reduced to two dimensions if we only plot the corners, or *breakpoints*, of the possibility distribution, as seen in figure 3.1.

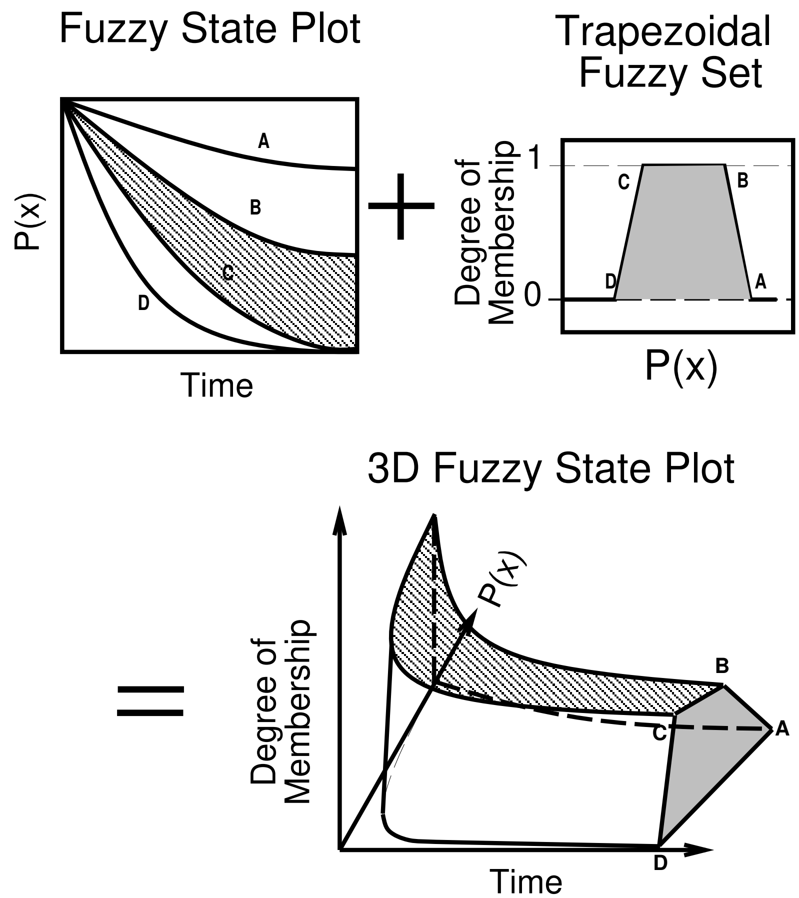

Figure 3.2 Output Format for a Fuzzy Markov Model.

3.2 Requirements for the Model

There are several important requirements that the fuzzy Markov model must fulfill. The most obvious of these is that it must be better in some way than the crisp (standard) Markov model. This requirement is expected to be met by the fuzzy nature of the model. Chapters 3 and 2 contain arguments as to why a fuzzy representation is better than a crisp one for reliability analysis. As long as our new fuzzy Markov model gives a meaningful fuzzy output we will consider this requirement met.

The major benefit of using fuzzy reliability models is that they preserve the uncertainty accurately and reliably throughout the calculation. The uncertainty in the input should propagate through the model in a mathematical way so that the output uncertainty is correctly determined. A certain amount of 'pessimistic' error that overestimates the possibility of error is permissible, assuming it cannot be avoided or makes the calculations sufficiently easier. 'Optimistic' error that underestimates the error possibility is less forgivable and should only be tolerated if it is smaller in magnitude than the inherent noise in the model. This requirement will be referred to as the *uncertainty criterion* for the rest of this thesis.

Another important factor to consider is complexity. The fuzzy Markov model is likely to be more complex than a crisp Markov model as it uses fuzzy possibility densities where the crisp model uses crisp probabilities. For the most commonly used type of Markov model, these probabilities are constants, allowing for minimal complexity. For fuzzy models using trapezoidal membership functions, we will have to deal four values for every one one in the crisp model. Interaction between these values can raise the potential complexity further. In the worst case, the complete fuzzy possibility set will have to be considered instead of just these four points. This is likely to increase the complexity of the model severely, if it is required. This requirement will be referred to as the *complexity criterion* for the rest of this thesis.

The final criterion that any new fuzzy Markov Model will be judged on is some sort of 'niceness'. A model that gives illogical, unintuitive, or overly complex output is not likely to be a good model. Although it can be hard to precisely define 'niceness', it is usually not hard to achieve consensus that certain models are not 'nice'. Additionally, several mathematical 'niceness' criteria are obvious, resulting in tests that exclude a model from being nice.

The first of these, *fuzzy niceness*, test to see if the fuzzy output of the model is a 'nice' fuzzy set. For our purposes, a 'nice' fuzzy set has $H_\gamma \subseteq H_\beta, \forall \gamma \leq \beta$, where H_α is the α-cut of the fuzzy output. In other words, we require that the lower possibilities cover a larger interval than the higher possibilities at all times. Also, although it is not a strict criterion, we would prefer our fuzzy sets to be continuous and smooth, for ease of interpretation and further mathematical operation.

Another niceness criterion is *probabilistic niceness*. The main requirement here is that we don't ever have any possibility greater than zero of probabilities outside of the [0,1] interval. Since these probabilities are impossible, it is unacceptable to have nozero possibilities for them. However, we will relax the the probabilistic axiom 'the sum of all probabilities equals one', as for our fuzzy arithmetic this is only true in the sense of a 'fuzzy number close to one'.

3.3 Fuzzy Markov Modeling Through α-cuts

One possible fuzzification of the Markov model would use methods similar to the fuzzy fault tree, where it was sufficient to propagate the extremal values of the α-cut through the fault tree as if it was crisp, and take the resulting extremal points as the corresponding α-cut for the output possibility distribution.

Unfortunately, this method is not sufficient for a good fuzzy Markov model, as it is valid only for trivial systems. As a counterexample, consider the three-state system shown in figure 3.3.

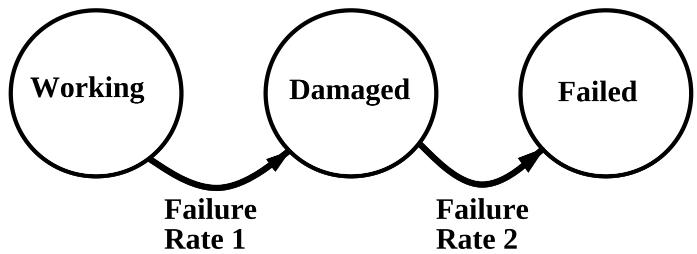

Figure 3.3 A Three State Markov Model.

Assuming the system is initially working, the population of the middle state ('Damaged') will initially increase over time, as the working population fails into this state. However, the rate of this state failing into 'Failed' state will eventually dominate, causing the this population to decrease towards zero. The time frame which this happens and general shape of the curve depends on the failure rates involved. When we propagate extremal α-cut values through this model, we quickly discover the problem shown in figure 3.3.

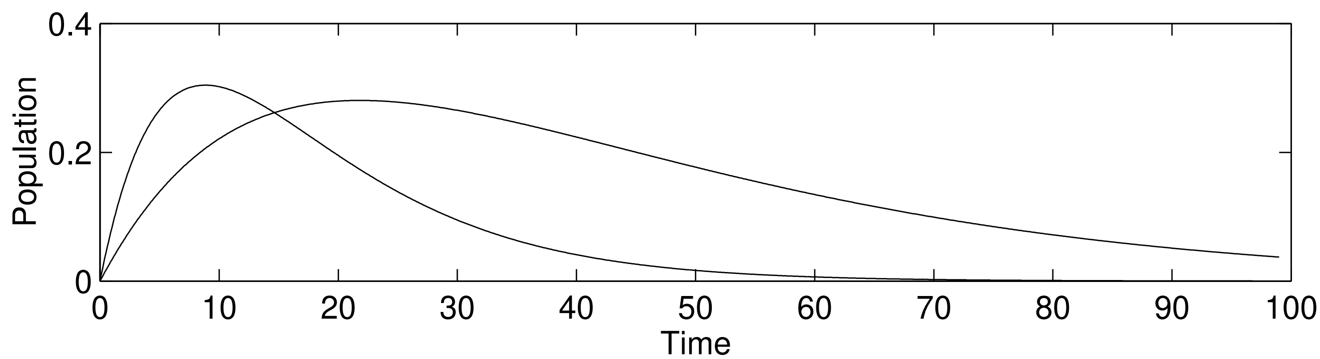

Figure 3.4 Extreme Values Fail to Produce a Valid Fuzzy Markov Model.

It is apparent that this method violates the uncertainty criterion described above. At the point that the two curves cross, the model says that there is no uncertainty in the population of the state. It can be shown, however, that this is not true if one considers the continuum of the possibility distribution, so this method is not a valid one for finding the fuzzy Markov model. An extension of this method is considered in a later section.

3.4 Fuzzy Markov Modeling Using the Extension Principle

The generalization of any crisp operation to a fuzzy operation on fuzzy numbers can be accomplished via the emphextension principle, as presented in Chapter 2. It is natural, then, to try to use the extension principle to fuzzify crisp Markov models. The model is simply solved as if it is crisp, using symbolic constants for the failure probabilities. The resulting equations are then fuzzified by substituting fuzzy possibilities for the probability constants and fuzzy operations for crisp ones.

Although theoretically promising, it was quickly determined that this approach violated the probabilistic niceness criterion - i.e. it resulted in nonzero possibilities for impossible probabilities. This is made clear by the following example. Consider the simple two-state Markov model example given in Chapter 2. If we look at is as a discrete model, we can set up the following arbitrary problem. Let the zero cut of the failure rate λ be the interval (0.3, 0.7), and zero cut for the population of the working state W be (0.1, 0.5). Let the corresponding cuts for the repair rate α and the failed population F be (0.2, 0.5) and (0.4, 0.8). On the next time step, the fuzzy arithmetic gives the working state zero cut as $W - W \times \lambda + F \times \alpha$. Using the fuzzy mathematical operations from table 2.1, we see that the new α-cut is $[0.1 - 0.5 \times 0.7 + 0.4 \times 0.2, 0.5 - 0.1 \times 0.3 + 0.8 \times 0.5] = [-0.17, 1.07]$. This cut allows for nonzero possibilities of impossible probabilities, which is clearly unreasonable. A typical result of this approach is seen in figure 3.4.

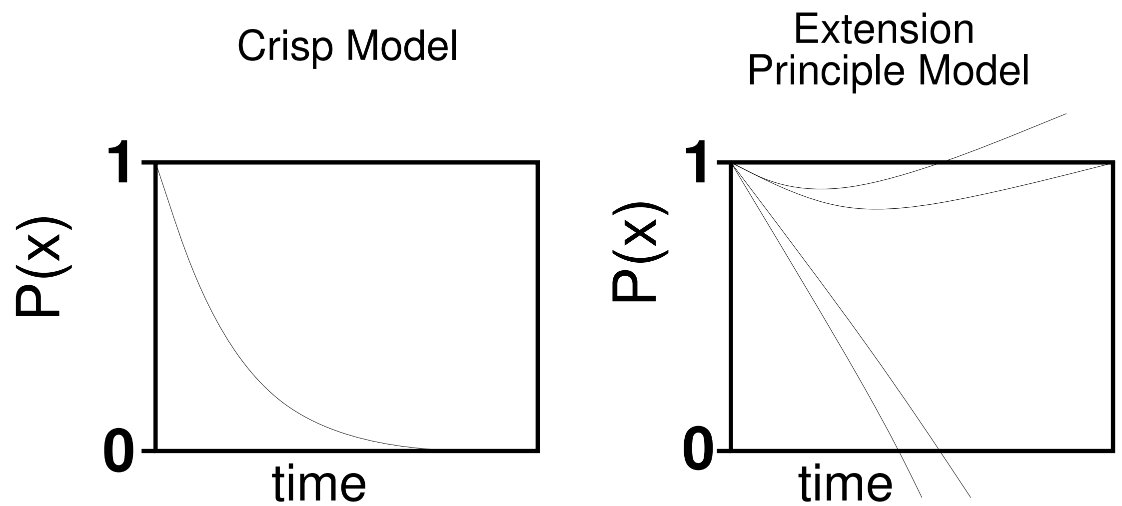

Figure 3.5 Extension Principle-Based Fuzzy Markov Model.

This is not a property of the discretization (The fuzzy zero cut for the continuous model is [0.24,1.85] after one time step with the rates given above.), but rather of the fuzzy math itself. It takes the worst case scenario of the lowest population, lowest repair rate and probability of being in the failed state with the highest failure rate and probability of being in the working state resulting in a probability of being in the state which is less than zero. This impossible situation is generated because the fuzzy arithmetic uses the most extreme *possible* probability in each stage of the calculation, not caring if different probabilities are used for the same value or if the probabilities in question don't add up to one.

This property is intrinsic to fuzzy arithmetic, and very useful in other applications such as fault trees. The inherent problem is in the additivity axiom of probability, and in fault trees, the failure and state probabilities never added or subtracted with other fuzzy numbers. Modification of fuzzy mathematics to force compliance with the additivity property of probability were difficult to usefully construct. All of the attempts made to do so resulted in logical self-contradiction, total loss of fuzziness, or unacceptable loss of information.

Therefore, despite the fact that the extension principle and fuzzy math are the ideal theoretical tools for fuzzification of Markov models, it seems unlikely that this approach will prove fruitful. It is not inconceivable that there is some clever way to overcome these problems, however, so this approach should not be completely ignored in future research.

3.5 Fuzzy Markov Modeling Using Fuzzy Integrals

As senn in Chapter 2, considerable work has recently been done in the field of fuzzy Markov modeling, using the concept of the fuzzy integral. It would be extremely convenient if this work could be adapted to reliability. Unfortunately, this is not the case.

The problem lies in the fuzzy integral. Although a fuzzy integral takes the fuzzy possibility of a fuzzy event, the result of such an integral is crisp! Although this may be a logical thing to do in some instances, it is not appropriate for the problem considered in this thesis. The uncertainty criterion requires that the uncertainty of the input of the model be reflected in the output of the model, and this is clearly not the case for the fuzzy integral, where the arguments are uncertain but the results are not. This approach bears a considerable resemblance to the probabilistic method of using the mean of the guessed failure rate as the 'official' failure rate, merely with a different method of finding the mean. It shares the same problems as well.

Note that this thesis does not claim that the fuzzy integral Markov model is not useful, merely that it is not a solution to the problem being considered in it's current form. There are several demonstrated applications for this kind of model [16, 23, 33]. It is also quite possible that the fuzzy integral can be adapted to to produce reasonable fuzzy possibilities of fuzzy events as well as crisp ones, perhaps in conjunction with the extension principle. This is discussed further in Chapter 6.

3.6 Generation of Fuzzy Markov Models Through Close Sampling

When we examined fuzzy Markov modeling using α-cuts in section 3.3 we only considered the approach where we solved for the extremal values of the cut. It is natural to consider what would happen if we considered all of the values in between as well.

In effect, this approach is attacking the problem from first principles. If the failure rate is in a certain interval, we can determine the possible behavior of the system by examining the behavior of the models resulting from every possible value on this interval. Since it is not known which of these values is the correct one, one can simply consider them all! If this is a possible approach, the problem is solved.

Of course, this approach is not easy. Since an interval contains an infinite number of points, one needs an infinite number of Markov models to solve the problem. This is clearly impossible, but if one assumes enough smoothness in the model to guarantee close values of the failure rate will give similar curves in the output, one can reduce this to a close sampling of these values instead of a continuum. Areas on the population graph that are between different plots can be assumed to be covered by some probability value between the the values that resulted in those plots. See figure 3.6 for an example of how this can work (although the sampling for the figure is actually somewhat coarse for legibility purposes). Complexity for this approach is still high, but a solution the problem is now within reach.

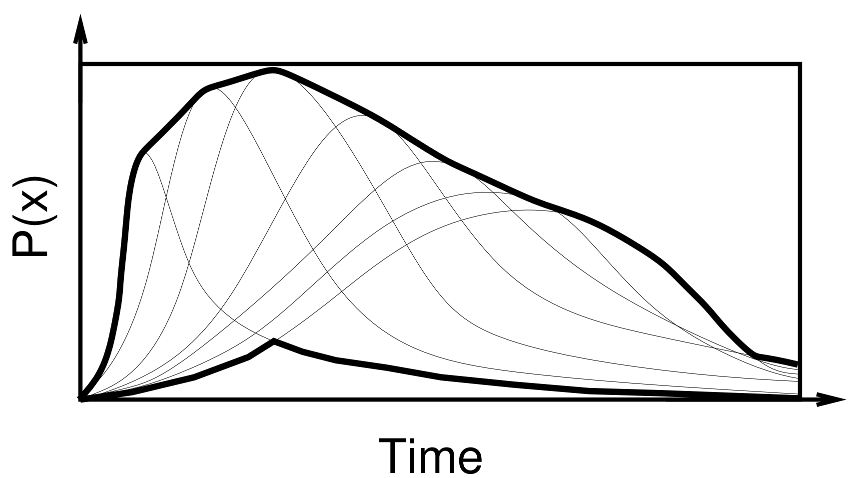

Figure 3.6 Fuzzy Markov Modeling Through Close Sampling Method.

Despite its brute force nature, this approach meets all of the requirements listed for the fuzzy Markov model at the beginning of this section except for one - complexity. Close sampling requires that many crisp Markov models be solved to solve a single fuzzy Markov model. If one is taking N samples on the interval, and there are M fuzzy failure rates, N^M crisp Markov models must be solved. It is easy to see how this can quickly grow to an unreasonable number of calculations.

The close sampling approach described above is the method that this thesis uses to calculate fuzzy Markov models. Despite the complexity issue, it is the only method found that has neither lost the important information nor resulted in impossible or useless output. Thus, the original problem of finding a fuzzy Markov model has evolved into the problem of simplifying and implementing the close sampling fuzzy Markov model.

In systems with many similar components in similar roles, this can be accomplished by grouping the failures of these components together in the Markov model. Instead of having a state representing 'pressure sensor 23 has failed', for example, we have 'a pressure sensor has failed'. Provided the failure of any single sensor has a similar effect on the system, this is a valid simplification. This often also allows us to use a single possibility distribution for all of the similar components, cutting down the number of crisp Markov models that need to be solved considerably.

A complex system with many different parts will probably have many fuzzy failure rates to deal with, more than enough to make a fuzzy Markov model impractical. However, when examining the failure characteristics of any complex system, we are quite likely to organize it into subsystems. This increases our understanding of the system, as the human mind is limited in the number of independent variables it can consider simultaneously. For example, if we were examining the falure characteristics of a robot arm, we might want to consider joint failures in our primary analysis. Once we knew those characteristics, we could then sharpen our focus to a model of the individual joints, considering motor, sensor, and mechanical failures. Then, if necessary, we can consider motor components, then subcomponents, and so on. This kind of simplification comes naturally to most and is helpful in promoting greater understanding of the system.

We can use the natural scheme of organization above to simplify our fuzzy Markov models. All we need to do is find a way to group the failure rates of the individual components into a single component failure rate. Fuzzy fault trees are ideal for this purpose. They are easy to implement, fuzzy mathematically sound, and specifically designed to determine failure rates for collections of components. Fuzzy Markov modeling using fuzzy fault trees for simplification promises to be a powerful reliability tool.

Chapter 4

The Fuzzy Markov Model Applied to a Test Problem

4.1 A Test Problem

The classical test problem in robotics is the two degree of freedom, planar manipulator. It is complex enough to be interesting (it has a large planar workspace), and simple enough to keep the mathematics down to a reasonable level. From a reliability engineering point of view, it is interesting to investigate the effects of redundant systems on this robot [38]. Kinematic redundancy arises when more degrees of freedom are available than are needed to perform the task. For the planar robot interested only in end-effector position, the required number of degrees of freedom is two. If a robot in this situation possesses three degrees of freedom, it can still reach a significant fraction of its workspace if one of the joints is frozen. Sensor redundancy occurs when there is more than one sensor at each joint, allowing sensor failures without joint failures. Actuator redundancy is similar, but much harder to implement in practice and will not be considered here. This chapter will examine four distinct robots: the non-redundant robot with two joints and one sensor per joint, the partially redundant robots with just sensor or kinematic redundancy, and the fully redundant robot with both sensor and kinematic redundancy. The problem is to determine how much more reliable the redundant robots are. The extra components will increase the incidence of component failure, while it is not obvious a priori that the redundancy will have sufficient beneficial effect to balance this. Fuzzy fault trees and fuzzy Markov models will be used to examine these effects.

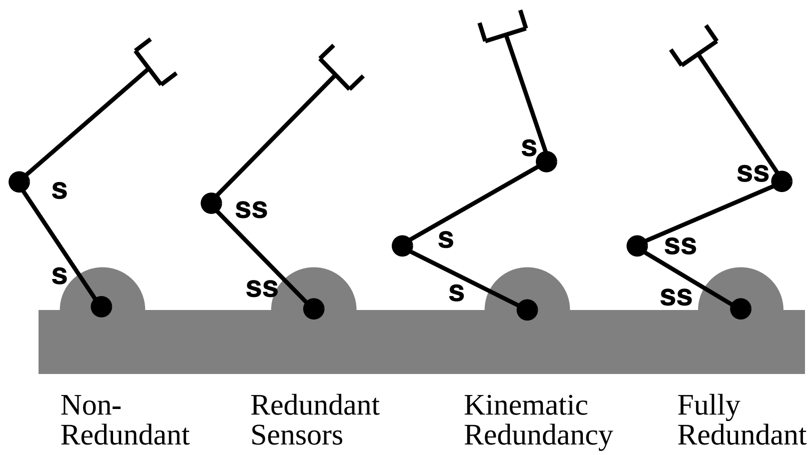

Figure 4.1 Test Manipulators.

This thesis uses the part failure rates given in the 1995 Nonelectric Parts Reliability Data (NPRD-95) failure rate tables [9] for electric motors and optical encoder sensors. This work notes that 68 percent of observed failure rates will be between 0.22 and 4.5 times the given value, and that 90 percent will be between 0.08 and 11.9 times the reported value. Following the example set in Walker et. al. [38], this thesis takes these values as the one α-cut (one-cut) and zero-cut, respectively, of a trapezoidal possibility function representing the fuzzy failure rate of the component. Note that the time unit for these rates is 1000 hours of operation.

Components	Reported Value	Zero-Cut	One-Cut
Electric Motor	0.00924	[0.000739,0.11]	[0.00203,0.0416]
Optical Encoder Sensor	0.0155	[0.00124,0.184]	[0.00341,0.0698]

Table 4.1 Fuzzy Component Failure Rate α-cuts.

4.2 Fault Tree Analysis of the Example Robots

Fuzzy fault trees of the example robots are given in appendix A. These trees assume that a robot fails if it has less than two remaining working joints, and that a joint fails if the motor fails or if all the sensors on the joint fail. As noted in chapter 2, fault trees are not suited for examination of partial failures.

If we solve these fault trees both crisply and fuzzily according to the methods described in Chapter 2, we get the results shown in table 4.2. These results reresent the failure rate of the entire robot over 1000 hours of operation. The zero-cut and one-cut columns describe a fuzzy trapezoidal possibility function for the failure rate, the crisp column gives the results of a crisp evaluation of the respective fault trees.

Note the extra information that the fuzzy treatment makes available, once the entire range of possible failure probabilities appropriate to the given fuzzy failure rates is used. The uncertainty in the failure rates of the components is directly

Configurations	Zero-Cut	One-Cut	Crisp
Non-Redundant	[0.00395,0.473]	[0.0108,0.205]	0.0486
Redundant Sensors	[0.00148,0.260]	[0.00408,0.0940]	0.0189
Kinematically Redundant	[0.0000117,0.208]	[0.0000886,0.0349]	0.00181
Fully Redundant	[0.00000165,0.0577]	[0.0000125,0.00640]	0.000270

Table 4.2 Fuzzy Fault Tree α-cuts.

visible in the uncertainty of the system failure rates. These calculations also indicate that kinematic redundancy is roughly equal to sensor redundancy for high component failure rates, but vastly superior if the failure rates are low. Note that if only the 'crisp' column was available, the observation above would have been missed, even if the failure rates had been listed with statistical uncertainties. This example shows some of the power of fuzzy arithmetic when applied to reliability analysis.

4.3 Fuzzy Markov Model Analysis of the Example Robots

For the robots we are considering, the Markov models become complicated quickly. Even the simple non-redundant robot has four separate components of interest (one sensor and one motor per joint), each of which can be either working or failed. This leads to 2^4, or 16 possible states of the system. The fully redundant robot has 9 components, leading to 512 possible states. This is too many states to deal with effectively. It is necessary to group and cut states to simplify until a reasonable number is reached. One approach, as presented in chapter 3, is grouping the components. For these robots, three categories of failure are sufficient: motors, single sensors, and pairs of sensors on the same joint. The fuzzy failure rates are similarly grouped into sensor and motor failure rates. Pairs of sensor failures are derived from successive single sensor failures. This approach remains valid as long as similar components are used in all three joints.

In the Markov models for these robots presented in appendix B, states with one or more of these parts failed are represented by states with labels **M**, **S**, and **P** respectively. States are characterized only by how many motors, sensors, and matching pairs of sensors are failed. Another reduction is accomplished by lumping all system and joint failures which have the same cause together, regardless of extraneous subsystem failures. For example, if the fully redundant robot fails from a working state into the state where two motors have failed, additional failures are ignored, and the state is referred to as '**MM+**', where the '+' indicates that there may be other failed components. The final reduction of the model was to make the standard Markov simplification and assume the sensor and motor failure rates were constant across the robot and time. This is not a completely realistic assumption (burn-in and wear-out are neglected, as is the effect of specific location of the component on the robot), but it makes the calculations considerably easier to deal with and shows the methodology more clearly.

Another important simplification method is the simplification through fuzzy fault trees. This method is most useful for more complex systems, so only the fully redundant robot will be considered in this example. One obvious scheme for grouping failures is by joint, so fuzzy fault trees must be solved for each joint's components. Since this robot can continue operating with one failed joint, fault trees for two joints at a time must be solved for this case. States with a failed joint are represented by a **J#**, and failure rates are represented by a **F#**, where the # represents the index of the joint that failed. The fault trees are found in appendix A, while the Markov model is found in appendix B.

Note that the fuzzy fault tree method allows for all of the sensors and motors to have different failure rate distributions without requiring any increase in complexity of the model. This is a significant advantage. Fault tree modeling also allows for stress effects. In the case of this model, it is assumed that once one joint is failed, there is more stress on the other joints, and thus their failure possibilities can be increased. Again, it is possible to do this without increasing the complexity of the model.

4.4 Algorithm Used in Fuzzy Markov Calculations

As mentioned previously, a discrete model was chosen to implement the fuzzy Markov model. Modern robotic controllers are usually digital computers working in discrete time, so this approach is not unjustified. For the first four models, the sensor and motor failure rates from NPRD-95 [9], as fuzzified above for the fuzzy fault tree, were used. Since part of the purpose of the last model was to illustrate the use of different fuzzy failure rates for each component, arbitrary fuzzy failure rates of the same order of magnitude and degree of fuzziness to the NPRD-95 data were chosen for this model.

The following algorithm was used to update the state populations:

1. All states are initialized with probability zero, except for the initial state, which has probability one. If no state is explicitly labeled as the initial state, then the working state (no failed components) is the initial state.

2. Calculate the transitions out of each state. If λ is the probability of a transition from state A to state B in unit time, δ is the size of the time step, and P is the probability of being in state A at time $t-1$, then the probability of the $A \rightarrow B$ transition is $\lambda\delta P$. Larger time steps and failure rates can be simulated by using $(1-e^{-\lambda\Delta})P$, where Δ is the size of the larger time step. This prevents transition

probabilities greater than one while minimizing distortion for smaller steps and rates. This modification is necessary in the model as the zero-cuts derived from NPRD-95 spread over two orders of magnitude. The exponential was chosen here because the natural behavior of the system is exponential. Note also that this method fails to account for the effects for more than one transition being possible out of a state. It relys on sufficiently fine discretization to keep error at a minimum.

3. The sum of all the transitions out the state is subtracted from the sum of all transitions into the state. The result is added to the probability of being in the state at time $t-1$ to get the probability of being in the state at time t.

4. Repeat the previous two steps for each state for each time step until finished.

In addition to the states mentioned, the utility of the robot was calculated. The utility in a Markov model does not in general correspond to any one state of the model, but instead is a weighted sum of several states. For our examples, all states where all the robot joints are working is weighted by one, and a state where two out of three joints are working is weighted by 0.5. (Note the ability to deal with degraded states.) The utility of the robot is the topic of great interest in regards to reliability, as it shows a measure of how useful the robot is expected to be over time. It results in a numerical value of usefulness to compare different robot configurations.

4.5 Interpretation of the Fuzzy Markov Model Results

The first and simplest model is the model of the non-redundant robot as seen in appendix B and appendix C. Several things can be learned from this model. The initial/working state in this robot has the highest population of all the robots. This is not surprising, as this robot has the fewest components (two motors, two sensors) and thus there are fewer parts to fail. However, all part failures lead instantly to system failure, so the robot has the lowest *utility* of all the robots as well.

The lower bounds of the fuzzy sets are much lower than the upper bounds. For both the **M** and **S** states, the one α-cut alone covers most of the range of possibilities. This indicates that it is not possible to isolate one or the other failure mode as being predominant with the given data.

The fuzzy sets give a somewhat misleading impression of the error possibilities when considered together. Both states have high memberships in high probabilities at the same time. This does *not* allow both states to have very high probabilities or very low probabilities at the same time, as the axioms of probability would not allow this. Instead, these membership functions indicate that it is highly *possible* that either type of failure could be that probable, and that with the fuzzy probabilities we have, this *range* of probabilities is all we really know.

The model of the manipulator with redundant sensors, as seen in appendices B and C, tells us many things. Although the robot fails out of the initial state more quickly than the previous robot, it has a higher utility. There are six components in this failure model, so failures are more common. However, many of these are not fatal to the system. For example, four of the possible failures from the initial state are into state **S**, which is a working state with utility one. Note that the lower bounds of the utility are not significantly better than those for the non-redundant robot. This robot has a ‘weakness’ - high motor failure rates bypass its redundancy.

This model, unlike the previous one, has *transitory states*, or states that have no population in both the initial (new) and final (failed) states of the robot. The states **S** and **SS** are very possible for a large set of probabilities and times. This is because the time at which these functions hit a maximum is highly dependent on the failure rates. For high rates, this peak is very early, and for low rates, it happens very late (see figure 3.3 for an example). The positions in between are filled by various intermediate failure rates. Similarly, the lower bounds for these states are very low, as the low failure rates grow very slowly, and before they get too large, the high failure rates have already peaked and soon drop below them. Thus the wide range of possibilities for these states.

Note that these transitory states also represent an opportunity to repair the robot failure with minimal downtime. For failure states the robot is ‘down’, and can do nothing while waiting for repair work to proceed. Since this can involve such time consuming tasks as parts ordering, this can be a significant time period. For transitory states, the robot can continue working until the repair is ready to be made. For this model, it is possible to go directly from the working state to a failed state, but it is less likely than for the non-redundant robot.

Several anomalies can be observed on the transitory states. Notably, a small spike at the beginning of the state and a series of bumps alon the top edge of the state. These are both minor artifacts of the discrete model used, and have little effect on the gross characteristics of the states. It would take too much computing power to get rid of these anomalies using our method, especially considering the minor effect they have on the data.

The model of the kinematically redundant robot (same appendices) shows us some serious flaws that weren't visible from the fault tree approach, and also shows us an important benefit. Although the initial state decays approximately as fast as for the previous robot (both have six components), the utility of this robot is considerably lower. This is because of the effects of the lower utility of the partly degraded states. The fault tree model noted that none of the initial failures resulted in the failure of the robot. However, the fuzzy concept of utility used in the Markov model also incorporated the fact that *all* of the initial failures led to a state of decreased utility.

This robot does not have a weakness (working state failing directly into a failed state) in the way the previous robot did. The kinematic redundancy applies equally to sensor and motor failures immobilizing a joint. This means that the repair advantage of transitory states was more important for this robot, as all single component failures lead to transitory states. However, the higher number of components meant that very high failure rates had a stronger effect than on the non-redundant robot, so the utility lower bound is still low.

This robot, like those previous, also has very uncertain failure states. It is interesting to note the limited range of the **SM** failure state, however. This is the first state we encounter that requires both a sensor and a motor failure. This causes this state to be limited in its maximum population, as it is most populous when *both* failure rates are high, and thus the **MM** and **SS** failure rates are also high. This state also has the same upper bound on the one and zero cuts after a certain amout of time has passed. This is because the population of this state is influenced strongly by the *ratio* of the two failure rates. Note that this robot also has a relatively high lower bound on the one α-cut. This tells us that we can expect this failure to be important with a fair amout of certainty.

The model of the fully redundant robot lives up to our expectations of being the best robot in the test set. With nine components, this robot fails out of the initial state faster than any of the other robots. However, its utility is the highest by a wide margin, as it has protection from both kinds of component failure. Even the lower bounds show noticeable improvement.

The **PM** state is similar to th **SM** state in the previous robot in that it requires both types of failures to happen, and is thus limited to 0.5 in its upper bound for both the zero and one α-cuts. However, its lower bound is less prominent, as it requires a *pair* of sensor failures, which is less likely to happen than a single sensor failure. (Evidence for this can be seen by examining the populations of the various **P+** and **S+** states.)

The transient states **SM** and **SSM** exhibit similar behavior. Although these states are transient, they also contain both motor and sensor faults, and thus they exhibit the same ratio based convergence of α-cuts.

The large size of the states **M** and **S** is expected, as they result from the initial transitions out of the working state. However, the large possible population in the **SSP** state was something of a surprise. This can arise whenever the primary failure mode is sensor failure, but the rate of sensor failure is not too high. The **SSP** state has fewer working sensors than any other state, so failures out of this state would be at an unusually low rate.

The fuzzy Markov model simplified with fault trees, unsurprisingly, reveals different things than the previous models. Although the first joint seems to have a much greater possibility of failure than the other two, the fuzziness of the model shows that this isn't guaranteed, as the lower bound of this state's population is near zero. It also notes that it is possible that failure of the first joint could possibly dominate the failure characteristics completely, as it's maximum is near one. These points would have been missed in a crisp Markov model.

Compared to the previous fuzzy Markov model of this robot, this model has a higher population of the initial/working state. This is because failures that don't take out a joint are not considered in this model, while they were in the previous one. As a result, this model does not address the transient states of the robot in the same way, and is probably less appropriate for modeling for repair application.

This model has many fewer states than the previous one. It is much easier to understand and interpret. For more complex robots, this would be even more important. It is also possible to analyze the joint failures of this model in more detail, as a fuzzy Markov model, if this is required. This layering allows very complex systems to be dealt with.

Several themes can be found in the fuzzy Markov models above. The increase in the rate of component failure as reliability schemes are implemented is made clear. Higher reliability will paradoxically require us to deal with more component failures. However, those failures will be mitigated by the reduced rate of system failure that is evidenced by the higher utilities displayed by the fault-tolerant robots. Sensor redundancy provides a lot of this reliability for little effort, assuming that the motors are somewhat reliable. Kinematic redundancy adds a little more margin, but is not as useful, as the damaged robot is not as useful as the initial one. On the other hand, kinematic redundancy guarantees us that the first failure will not be fatal, and gives us improvement no matter which component has the higher fault rate, so it should be considered. Additionally, in terms of repair considerations, any modification that makes fatal failures into transient ones has definite benefits in applications where repair is an issue.

Complexity is also an important issue for these models. The number of states in the model increased along with the number of components. This was beginning to get unwieldy for the fully redundant robot. Simplification through fuzzy fault trees seems to be a reasonable approach to dealing with this issue, and has additional benefits as well.

Unfortunately, the most obvious fact learned from these models is that with data this fuzzy, we can't pin down many important facts about the system. The primary failure mode, for example, is not clear in any of these models - all we know is that the combined sensor and motor failure modes found on the kinematically redundant robots are limited in their upper range, or that certain failure have higer upper bounds than others. Still, the fuzzy Markov models have told us many interesting facts that we did not observe from the fault trees. It appears that this approach is worth further examination.

Chapter 5

Fuzzy Fault Tree and Markov Model Analysis of the MLDUA

5.1 The MLDUA

The Modified Light Duty Utility Arm, (MLDUA), is a robot arm designed to assist in the removal of hazardous radioactive waste from large underground storage tanks at Oak Ridge National Laboratory [6, 7, 31, 37]. The MLDUA is inserted through a narrow central access riser, and used to manipulate a 'hose management system' for waste extraction, as seen in figure 5.1.

The environment in these tanks is extremely hostile, and the waste involved is too hazardous to allow to escape. The MLDUA system has to meet many stringent safety requirements to deal with this problem [6]. In addition, the environment inside the tank is so hostile that the MLDUA itself is endangered. Extremely high radiation levels combine with explosive and corrosive chemicals to make the tank environment extremely dangerous to the robot. The overall effect of this environment on the robot cannot be predicted accurately before deployment. Stringent reliability requirements and uncertain failure characteristics thus combine to make the MLDUA system an ideal real world test case for fuzzy reliability analysis.

Considerable reliability work has already been done for the MLDUA. The design itself is very reliability conscious [31]. The robot has seven degrees of freedom, making

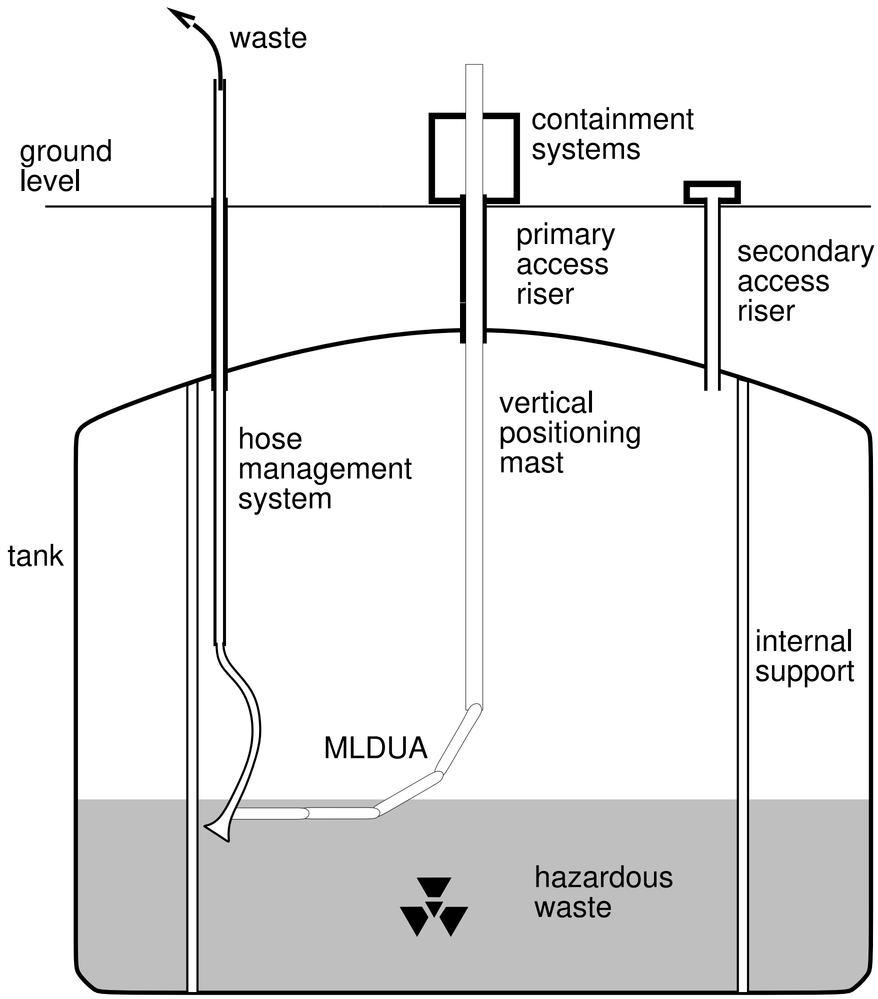

Figure 5.1 MLDUA Manipulator in Waste Tank.

it kinematically redundant. Each joint is monitored by two sensors. Five of the seven joints are powered by hydraulic motors connected to a 'limping system', which will be allow the robot to be straightened out and removed from the tank without power. This is an important consideration, as the robot arm is inserted through a narrow riser and must be 'limp' (straight) to remove from the tank. Due to the hazardousness of the tank's contents, only severely limited options are available for in tank repair if the system fails [31].

5.2 Fault Tree Analysis of MLDUA

An extensive fault tree analysis of the MLDUA system already exists [37]. This study considers the overall failure of the MLDUA system as well as tracking numerous lesser failures as subsidiary events. However, the resulting fault tree, which is reproduced in appendix D, is extremely large, and contains too many undefined events to be well suited to testing the ideas in this thesis.

Fortunately, many of the subsidiary events are sufficiently well defined and inter-related to be of use in the testing of fuzzy Markov models. These figures have been extracted from the overall tree, slightly modified to remove references to the rest of the tree, and placed in appendix D. The events of interest are component failures that lead to failure of the MLDUA while operating in the tank. Power system failure, joint failure, braking system failure, servo control failure, and limping system failure (all in appendix D) are all considered as separate events modeled by trees.

Table 5.2 gives the mean failure rates in failures per thousand hours of operation found in NPRD-95 [9] for the components of these fault trees. Note that these are fuzzified as shown in table 4.1 before use in the fault tree. Since this is a simple proportional operation, these values are not shown. Also, the frequencies of several events, such as pressure errors in the hydraulic system were not known at all. For these, a fuzzy representation of 'unknown' was used (zero and one cuts were [0,1]).

The one exception was the effectiveness of joint redundancy, which was assumed to always be effective for simplicity.

Fuzzifying these component failure rates and propagating them through the fuzzy fault trees gives fuzzy failure rates in failures per thousand hours of operation for the considered systems as shown in table 5.2. Note that some of these rates are greater than one. This is because the actual failure rates were calculated on a per hour basis, and scaled to the per thousand hour basis for consistancy with previous tables and simplicity of presentation.

Some of these failure rates are rather high, especially the joint failure rate. This is understandable, considering the large number of hydraulic motors in the joint fault tree and the high base failure rate of these components. However, we must also consider that the MLDUA will not be kept in the tank, but rather inserted in it for shorter periods, after which it will receive a thorough servicing [6, 31]. We also must consider the effect of the limping system on failures. Hopefully, most joint failures will be covered by this system. The order of failure events is important, however, as

Component	Failure Rate	Component	Failure Rate
Bearing	0.00291	Power Supply	0.0137
Electric Motor	0.0092	Rotary Joint	0.0075
Electronic Timer	0.0012	Sensor, General	0.00361
Hydraulic Motor	0.540	Sensor, Level, Liquid	0.0026
Hydraulic Pump	0.0470	Sensor, Pressure	0.00923
Hydraulic Valve	0.00882	Sensor, Temperature	0.00182
Mechanical Brake	0.1386	Strainer (filter)	0.00019
Optical Encoder	0.0155	-	-

Table 5.1 MLDUA Component Rates.

System Failure	Zero-Cut	One-Cut
Power	[0.0038, 0.7284]	[0.0105, 0.2755]
Joint	[0.2249, 33.0053]	[0.6184, 12.5873]
Servo Control	[0.0000, 0.00002]	[0.0000, 0.0000]
Brake	[0.0139, 2.0689]	[0.0383, 0.7825]
Limping	[0.0113, 1.6799]	[0.0311, 0.6356]
Joint (another joint damaged)	[0.1807, 36.3309]	[0.4967, 20.0215]
Servo Control (another joint damaged)	[0.0000, 0.00002]	[0.0000, 0.0000]

Table 5.2 MLDUA Fuzzy System Failure Rate α-cuts.

limping mechanism failure is not easy to detect, while joint failure is obvious, due to the sensors on the joints. Thus limping failure before joint failure can lead to a trapped MLDUA, while the opposite situation can be dealt with by removing the arm in the standard manner, using motors to straighten the arm. Since ordering is important, it is worthwhile to consider the system in the context of the fuzzy Markov model.

5.3 Fuzzy Markov Model Analysis of the MLDUA

As seen in the previous section, fuzzy Markov modeling of the MLDUA system is of interest to us due to the importance of the order of occurrence of some of the system failures. Two cases are considered. In the first, the operator runs the MLDUA for up to ten hours at a time, stopping only in case of failure. The second case considers a conservative operator who removes athe MLDUA shortly after any joint failure, in order to avoid a subsequent total failure combined with a limping failure, resulting in a trapped robot. The Markov model used for both of these cases can be seen in appendix D, with the 'c' failure rates only holding for the conservative operator.

The results of these two models, using failure rates derived from the fuzzy fault trees in the previous section, can also be seen in appendix D. Note that the time scale is much shorter than for the previous example's models (the units there were thousands of hours, these use single hours), and that the y-axis is logarithmic for all states but the initial one to make smaller probabilities visible. Note also that several of the states have lower bounds sufficient to keep them blow the lower edge of the graph, and are thus not visible. We can learn many things from these results.

The first thing one notices is the high possibility that the MLDUA won't make it through the average working day without some kind of failure. This is not good news, but it is not surprising, considering the complex nature of the system and hostile environment. This should be handled by careful maintenance, as is suggested in the technical reports on the MLDUA [6, 31].

One can also note that the possible probabilities for the 'trapped' state are fairly low for both Markov models, with worst-case values on the order of one in ten thousand. This may or may not be an acceptable risk level, depending on expected frequency of use and on the effectiveness of contingency plans for dealing with this failure.

It is also interesting to consider the fact that while the conservative operator decreases the chance of the trapped failure considerably (nearly half an order of magnitude), he doesn't remove it completely. This is due to the possibility of instant failures such as power or brake failure, which do not give the operator time to remove the robot arm. Note also that the non-conservative operator gets more work done, as he never voluntarily enters the failed state. This is clearly reflected in the model populations.

Chapter 6

Conclusions and Future Work

As new frontiers open up in robotics, fault tolerance and reliability of robots become ever more important. Robots are being used in new environments which are neither carefully controlled nor easily modeled. Robots are being sent into situations hazardous to them to keep humans out of danger. Both of these situations increase both the probability of failure and the uncertainty in its value.

Fuzzy Markov modeling, as presented in this thesis, is a viable technique for analyzing fault tolerant designs under the above constraints. It is complex enough to provide much useful information while maintaining the fuzziness inherent in the situation. It works well in conjunction with fuzzy fault trees, a well-established fuzzy reliability tool. Perhaps most importantly, it is not hard to understand, provided the underlying concepts of fuzzy sets and Markov models are understood.

The weakness of fuzzy Markov modeling by the method presented is it's computational complexity. The dimension of the model is linearly dependent on the number of fuzzy possibility distributions being considered. Currently, only trivial models, or models simplified by fuzzy fault trees or component grouping, are solvable in a reasonable amount of time.

Future work in the area of fuzzy Markov modeling is likely to focus on four areas. The first and most obvious of these is reduction of the computational complexity of the model. Similarly, further methods of simplification of the model should be considered. Additionally, Markov modeling is a very broad area, and this thesis only considers fuzzification of the most basic of Markov models. Expanding this technique to some of the modified Markov models shows promise. Fianally, application of this technique to several real systems is an important research issue.

The first problem, reduction of computational complexity, is very important to increasing the usefulness of these models. It is also unfortunately the most difficult. Although extension principle and fuzzy integral methods were shown to be inappropriate for fuzzy Markov models in their present form, there is a possibility that they can be modified in such a way that they would be. This research is likely to be intensely mathematical.

Further methods of simplification of the model is actually two different areas. The first is simplification of the Markov model. Research into this area should probably examine methods used to simplify crisp Markov models as a starting point. The second is simplification by reduction of the number of crisp Markov models needed in the fuzzification. Work into this approach is likely to be very case specific, but it is possible that general results could be obtained.

Repairable Markov models and time-dependent Markov models are of great interest in reliability [3]. The method developed in this thesis shows considerable promise to be applied to both of these.

Application of this technique to real systems is a long term project. Since it is applied during the design phase, results on its success and usefulness are not gained immediately, but instead suffer from considerable lag time. This avenue also depends strongly on the degree of industry collaboration with future research.

Appendix A

Fault Trees

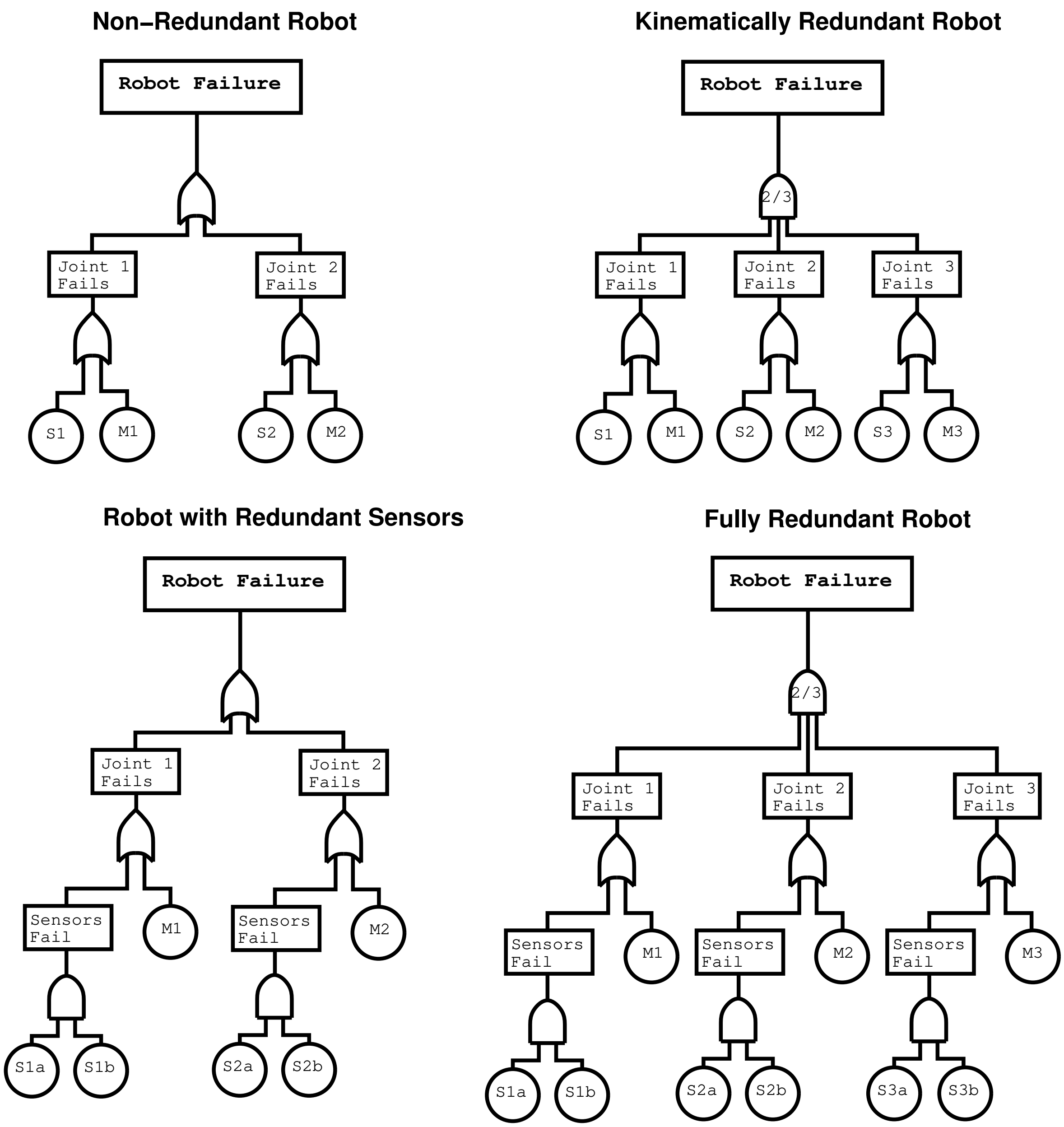

Figure A.1 Fault Trees for Example Robots.

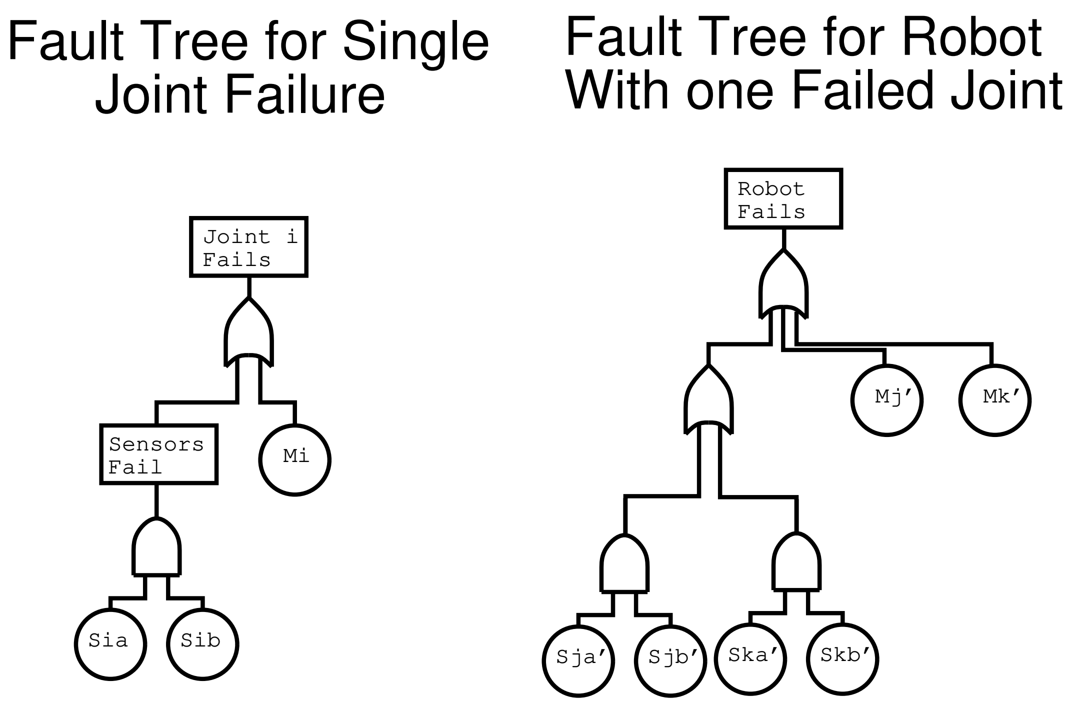

Figure A.2 Fault Trees Simplifying the Markov Model.

Appendix B

Markov Models

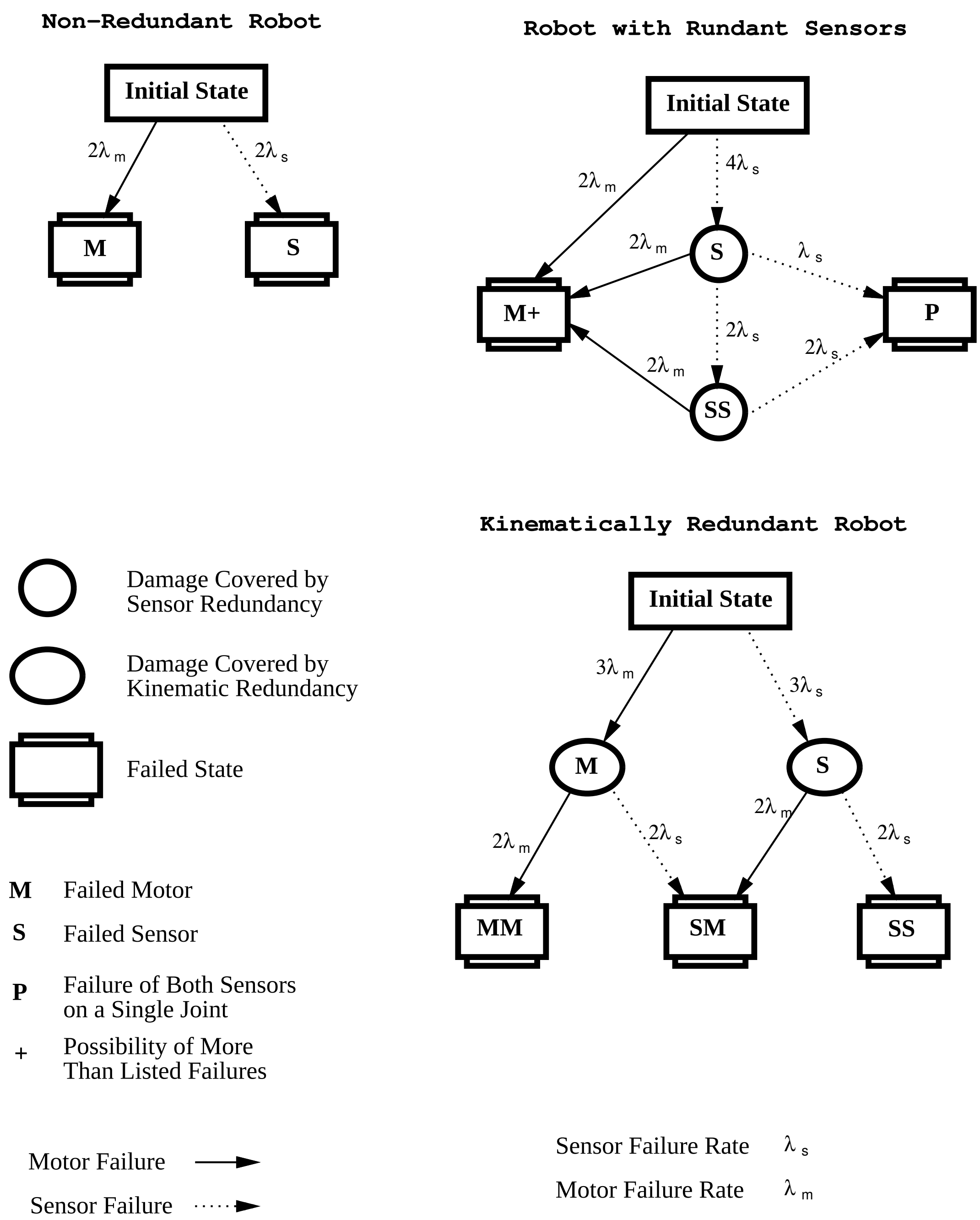

Figure B.1 Markov Models for First Three Example Robots.

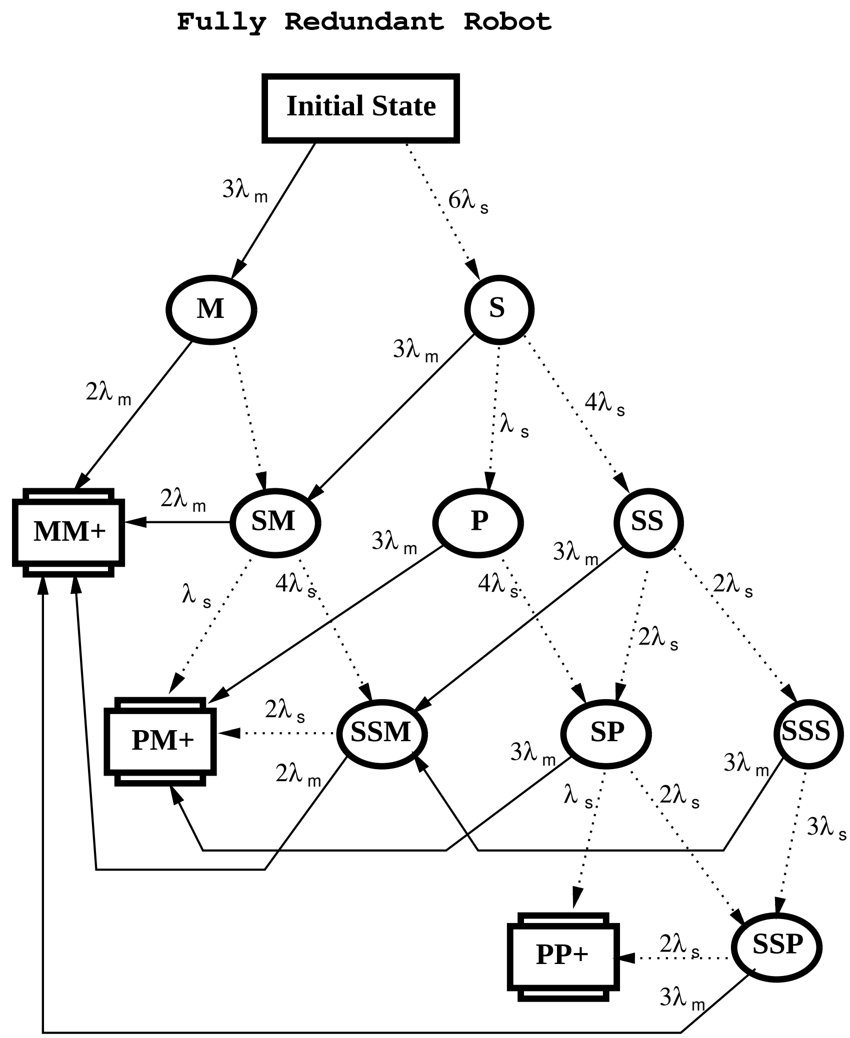

Figure B.2 Markov Model for Fully Redundant Example Robot.

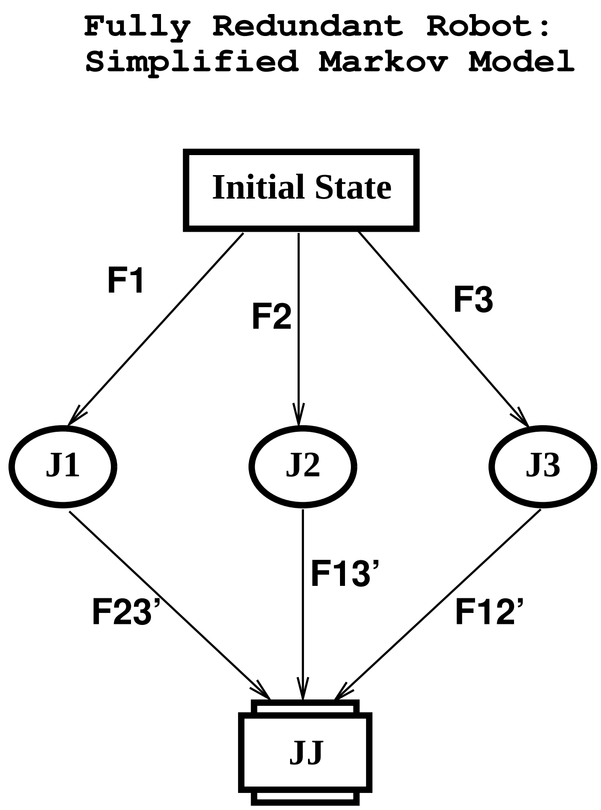

Figure B.3 Markov Model for Fully Redundant Robot, Simplified Using Fuzzy Fault Trees.

Appendix C

Fuzzy Markov Results

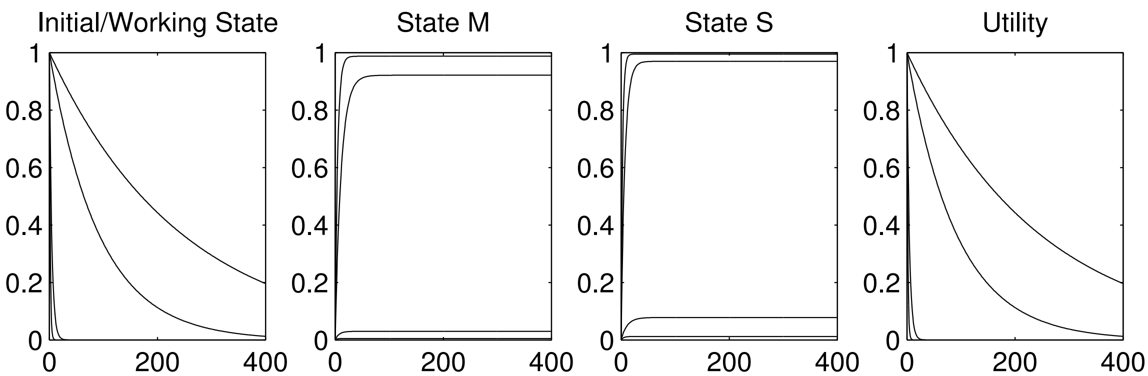

Figure C.1 Fuzzy Markov Output for Non-Redundant Robot.

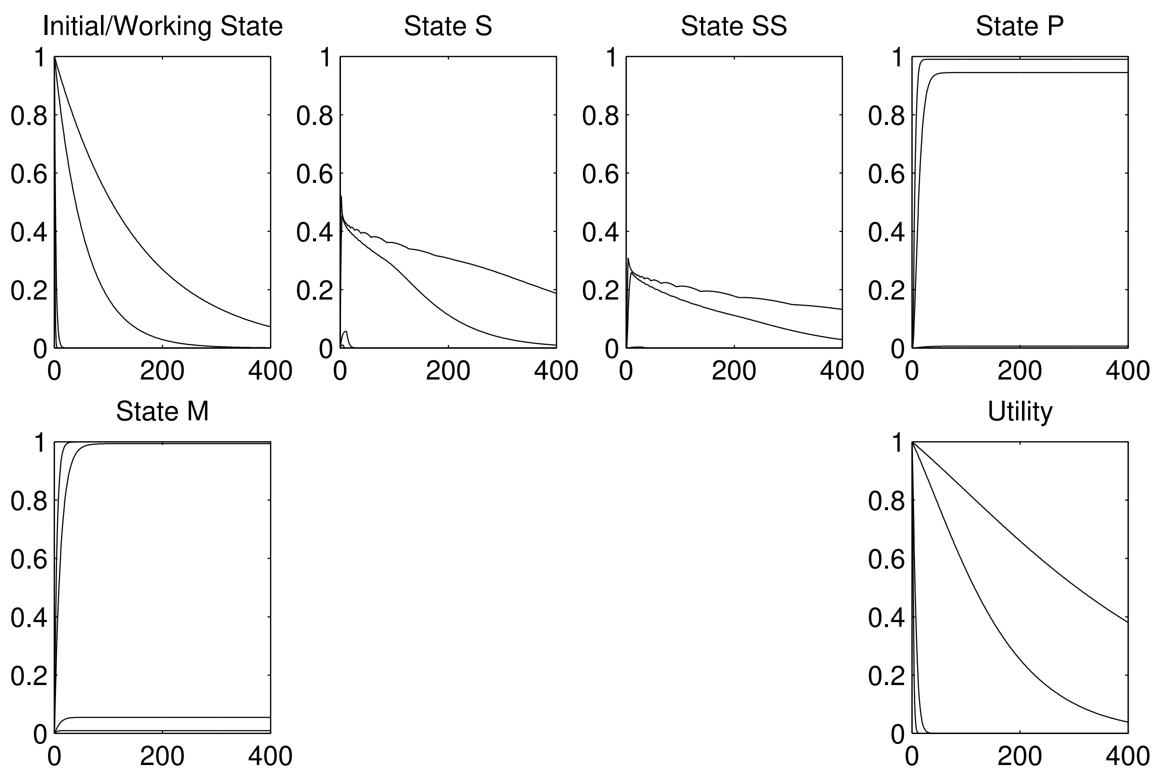

Figure C.2 Fuzzy Markov Output for Robot With Redundant Sensors.

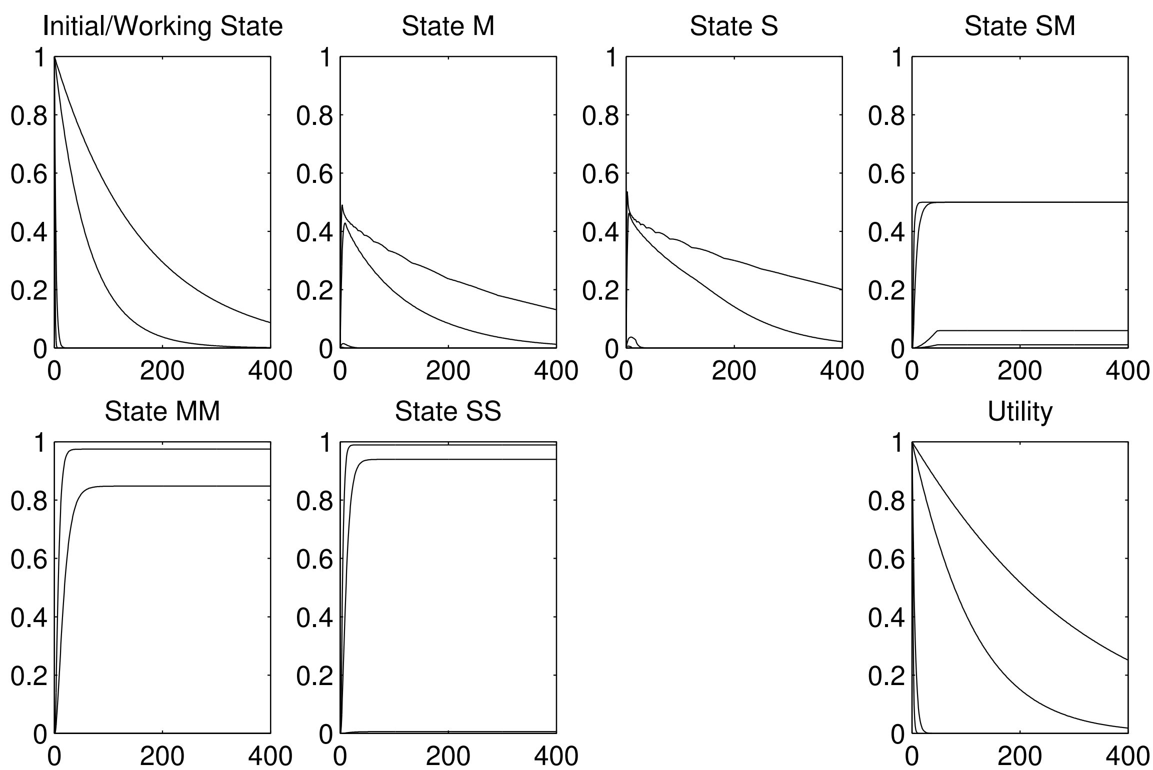

Figure C.3 Fuzzy Markov Output for Kinematically Redundant Robot.

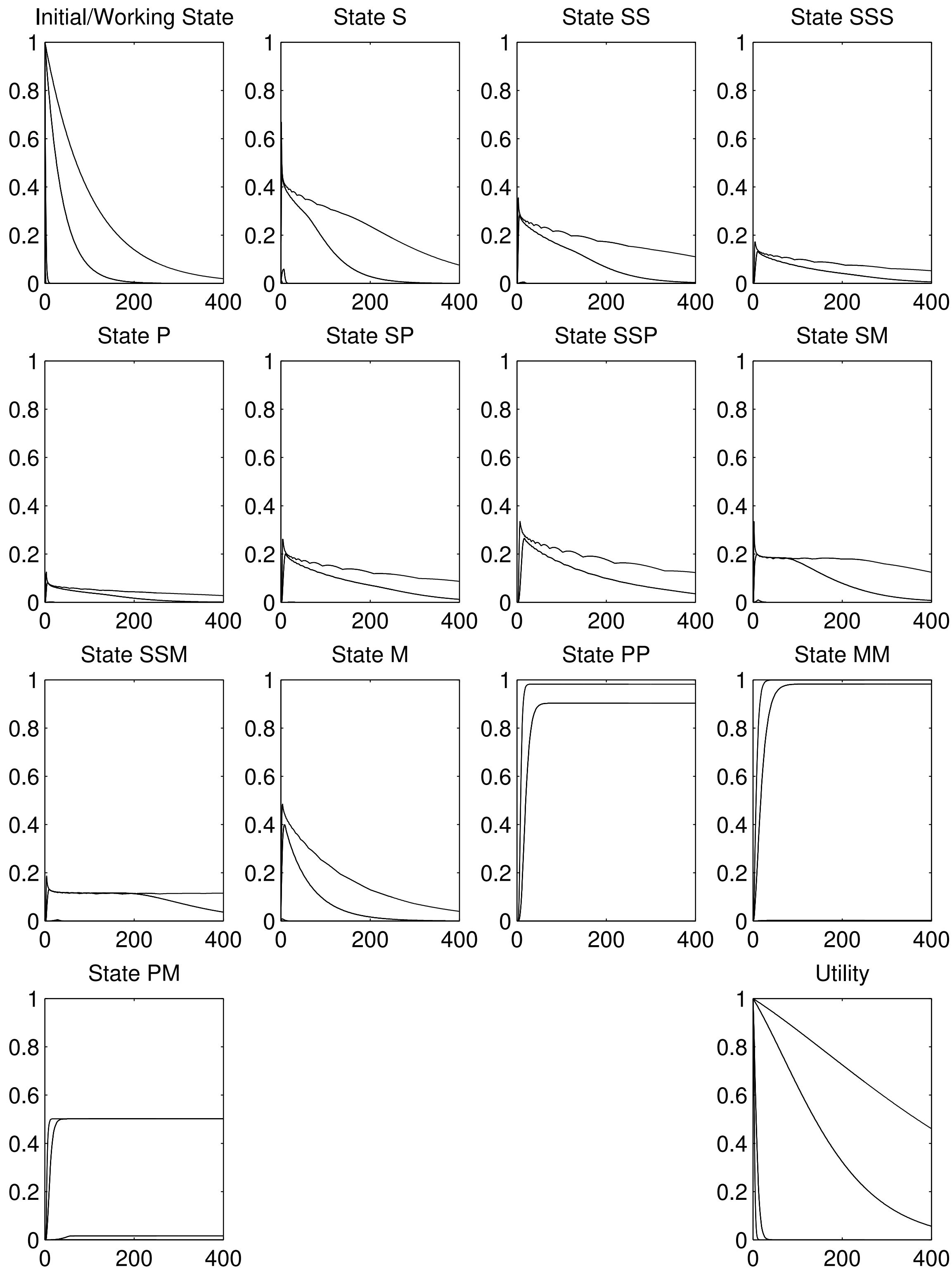

Figure C.4 Fuzzy Markov Output for Fully Redundant Robot.

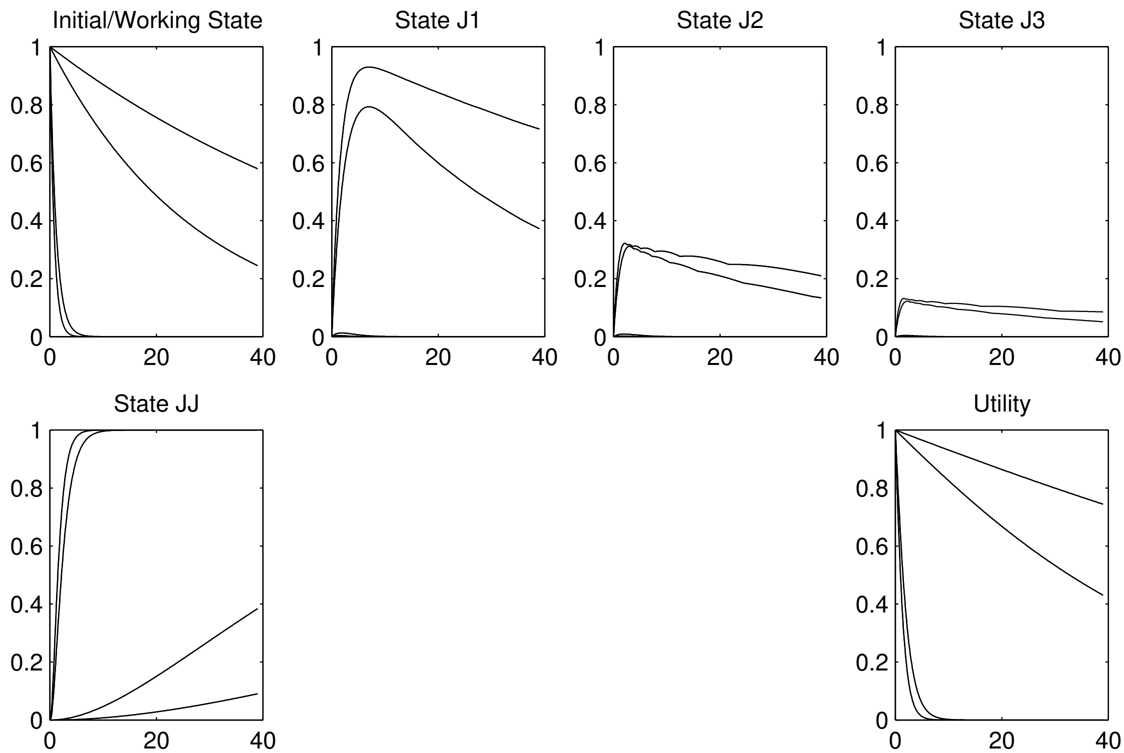

Figure C.5 Fuzzy Markov Output for Fully Redundant Robot Simplified Using Fuzzy Fault Trees.

Appendix D

Fuzzy MLDUA Reliability Analysis

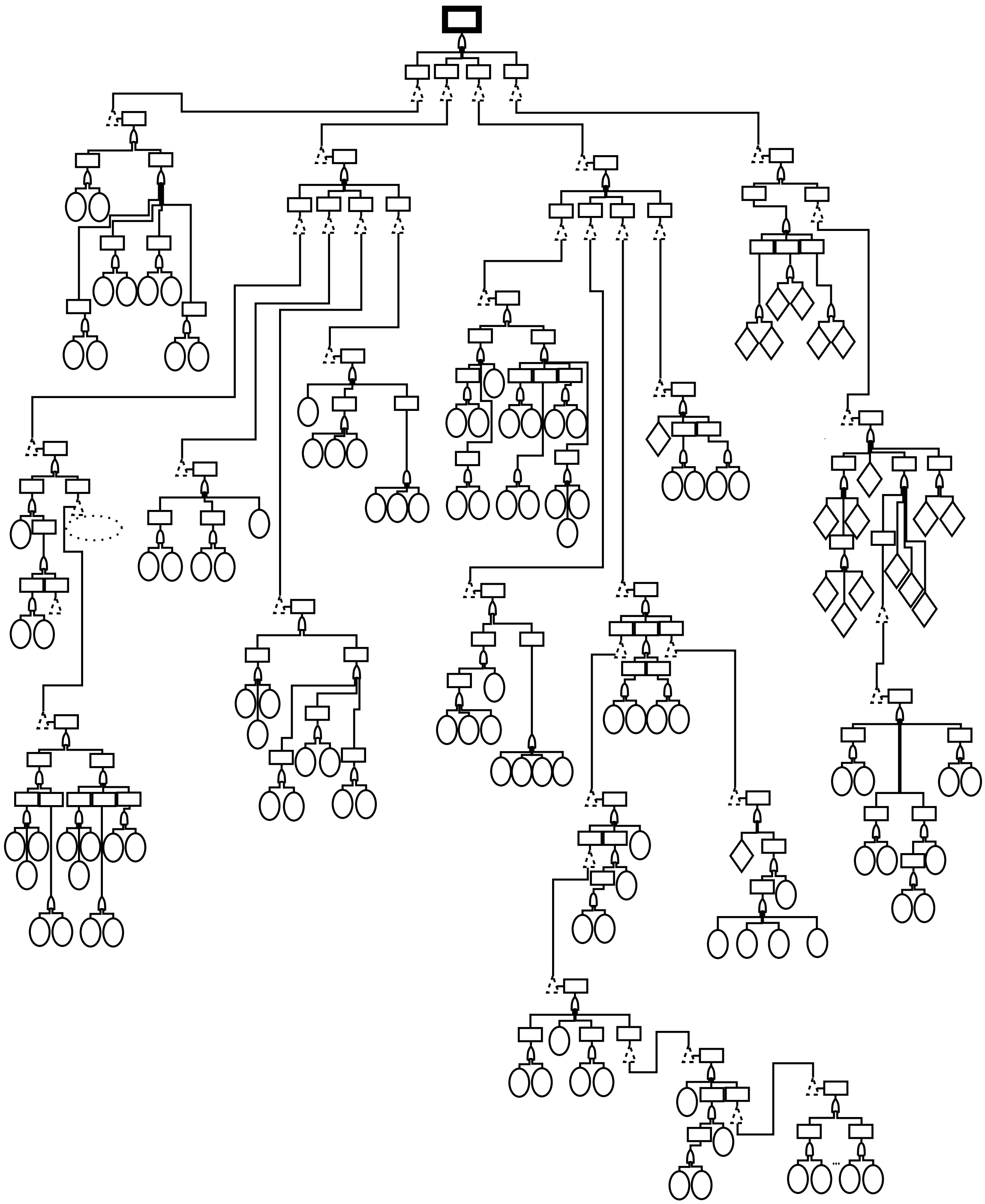

Figure D.1 Full MLDUA System Fault Tree.

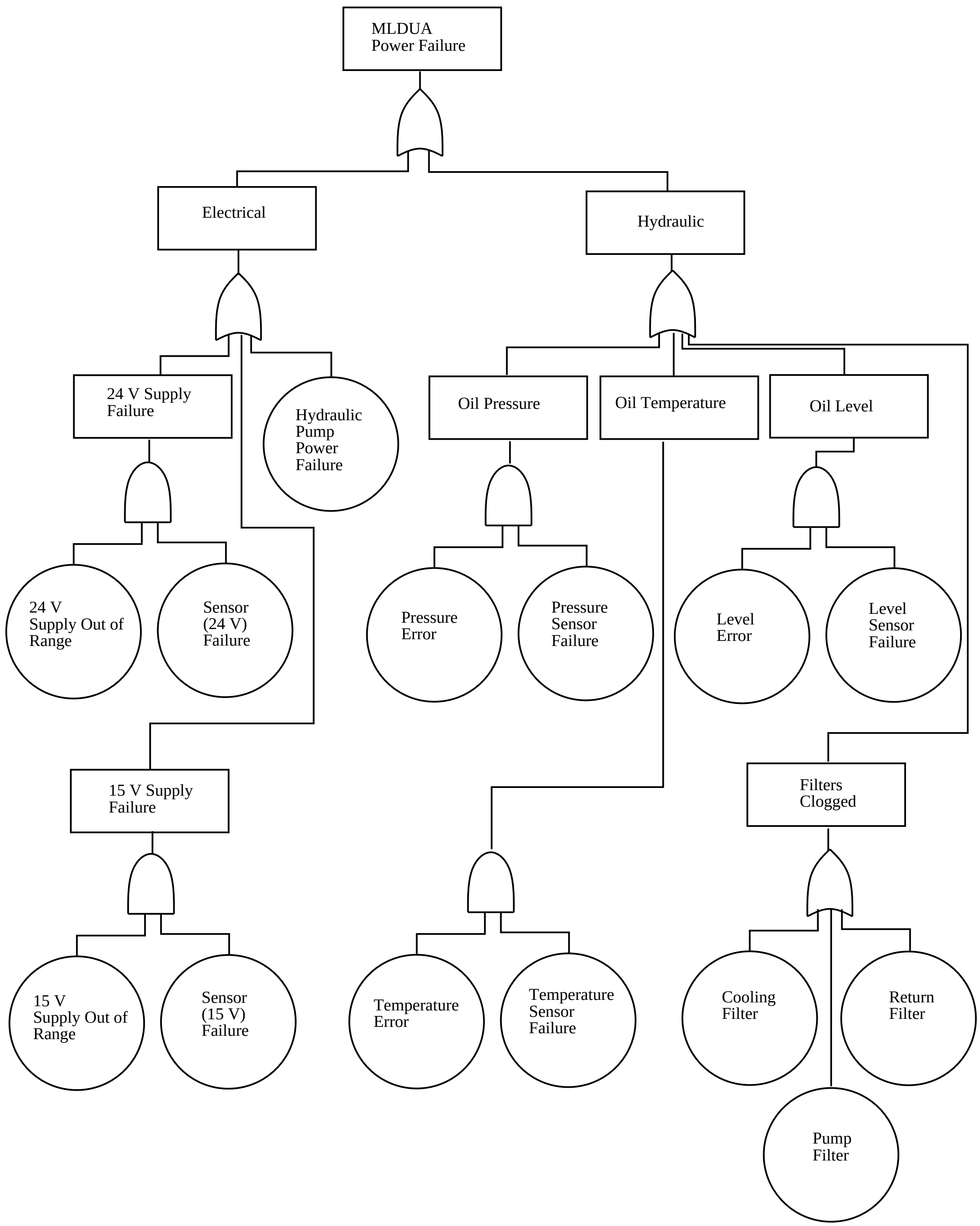

Figure D.2 MLDUA Power System Fault Tree.

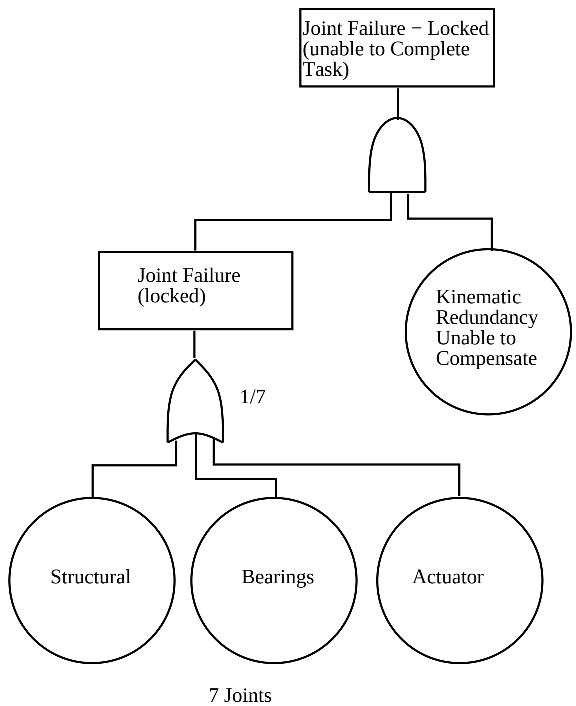

Figure D.3 MLDUA Joint Fault Tree.

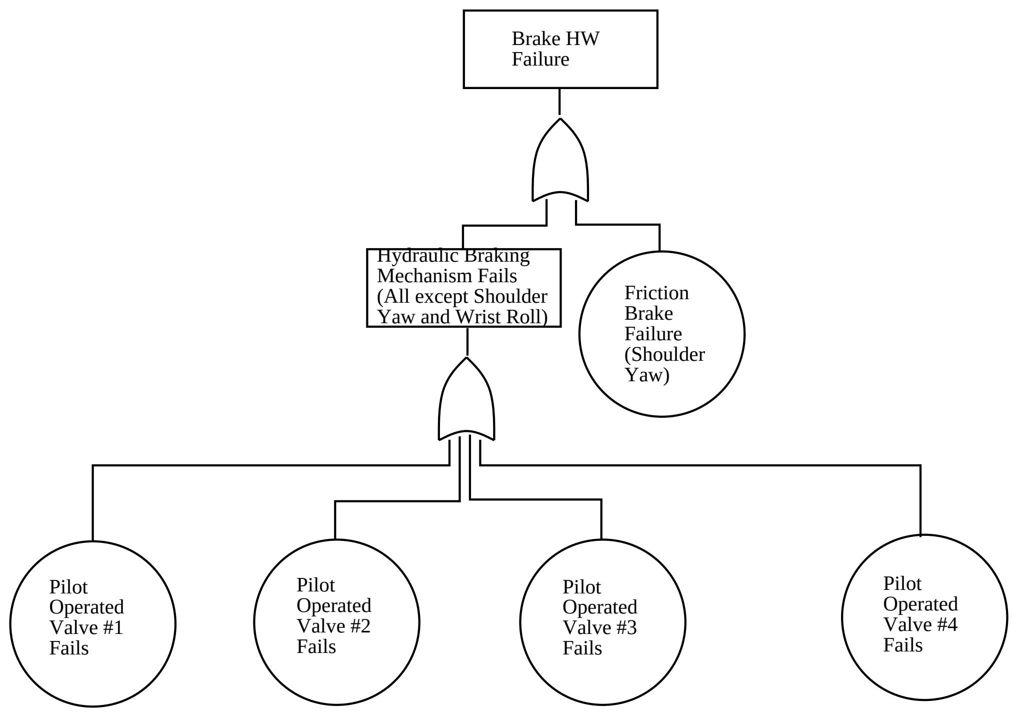

Figure D.4 MLDUA Brake System Fault Tree.

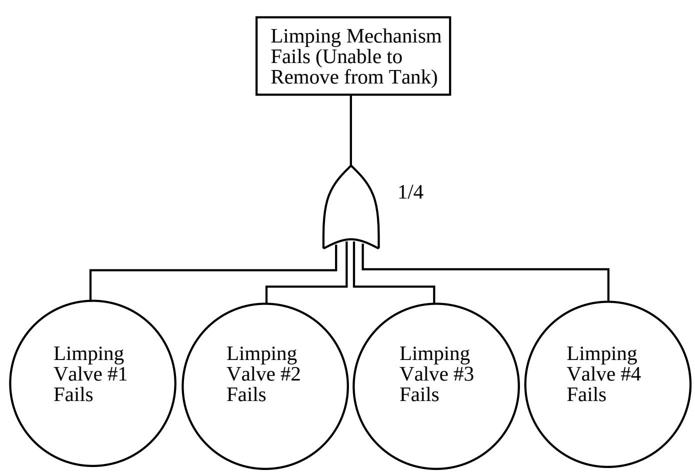

Figure D.5 MLDUA Limping System Fault Tree.

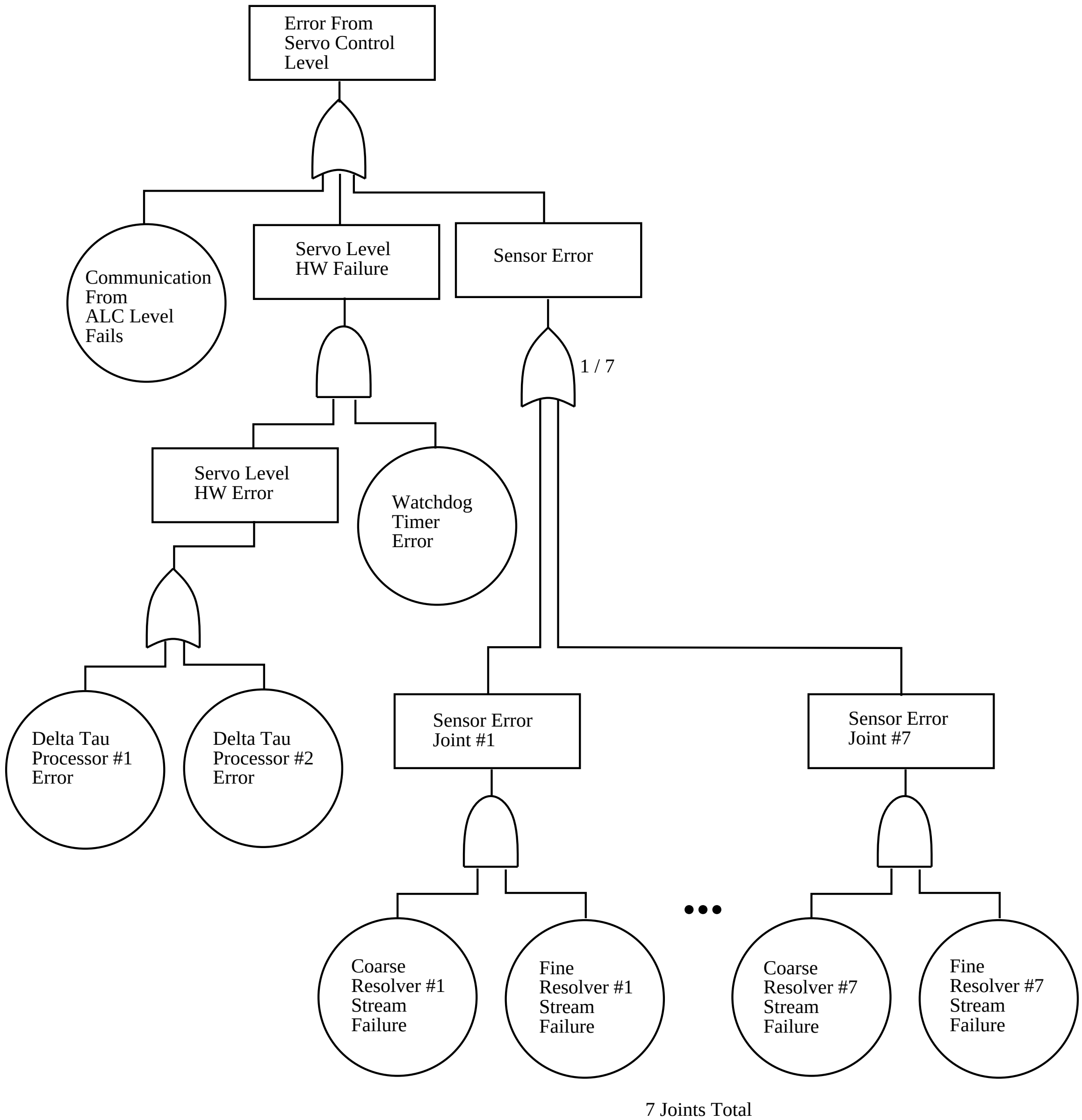

Figure D.6 MLDUA Servo Control System Fault Tree.

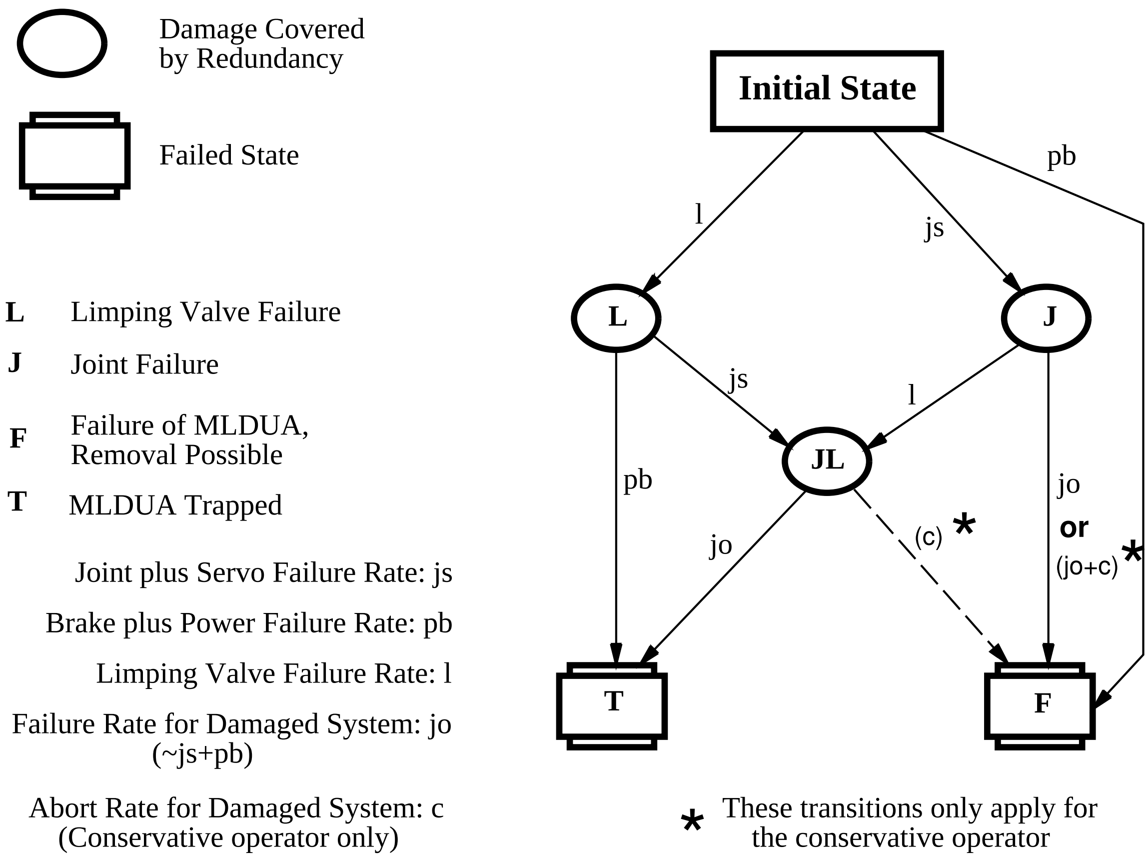

Figure D.7 MLDUA Simplified Markov Model.

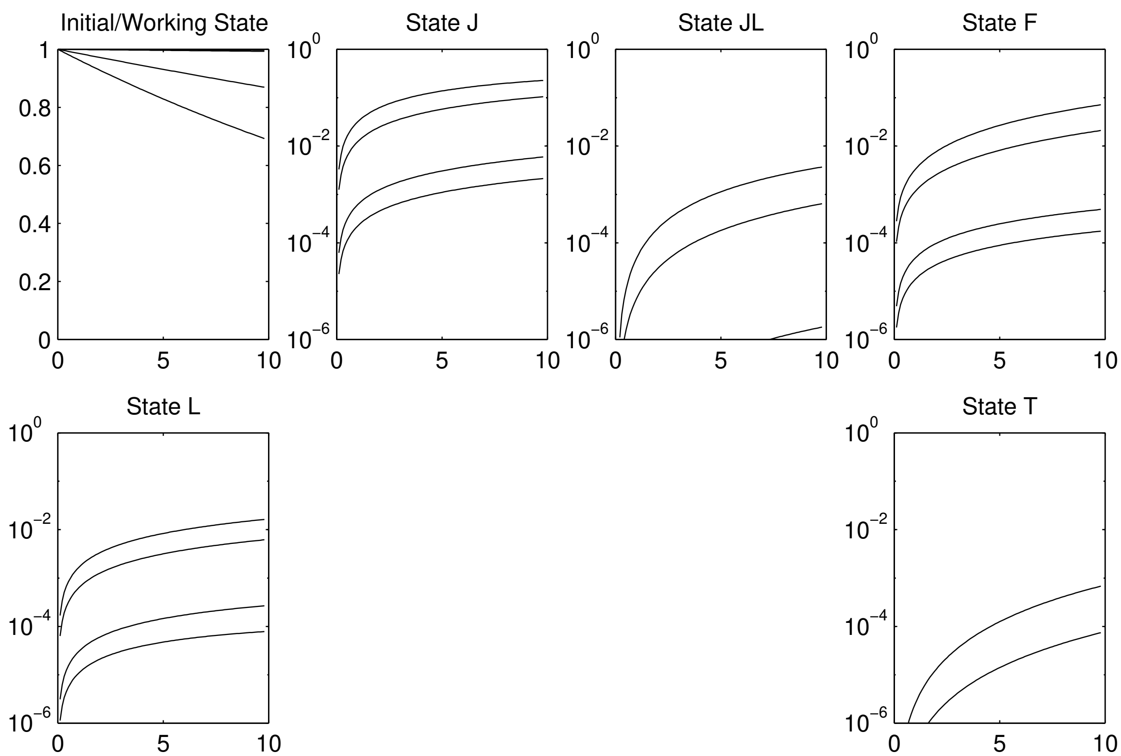

Figure D.8 MLDUA Fuzzy Markov Model Output.

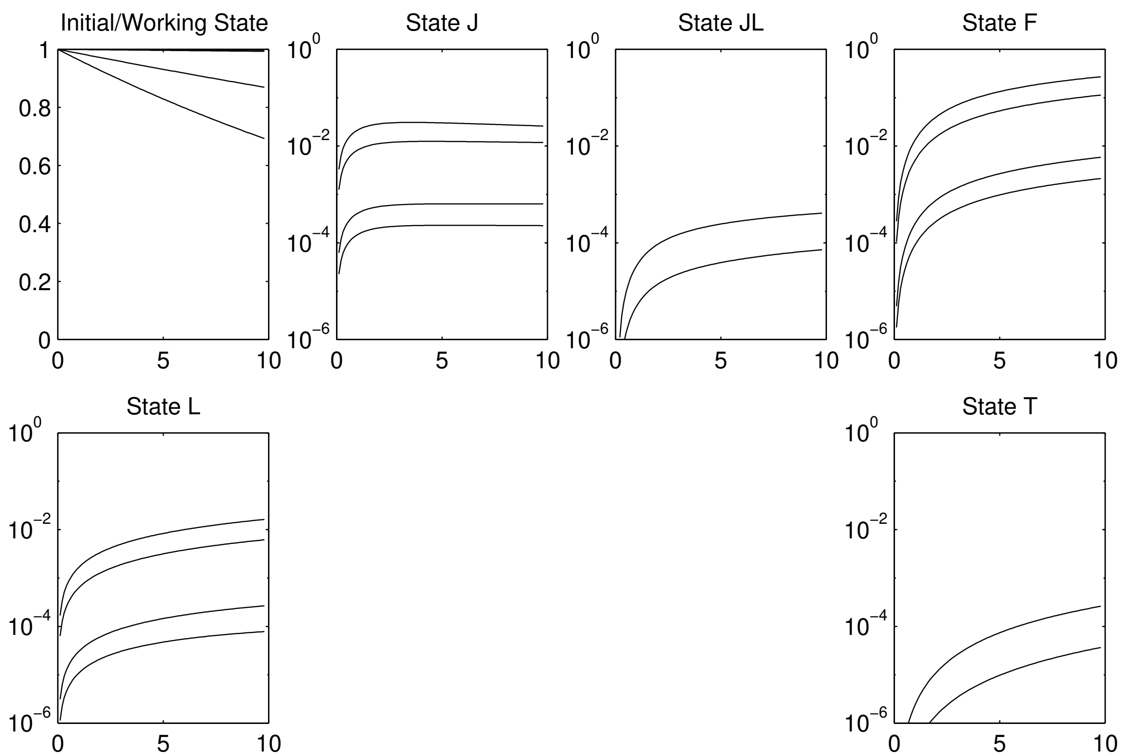

Figure D.9 MLDUA Conservative Fuzzy Markov Model Output.

Bibliography

[1] Marius Bâzu. A Combined Fuzzy-Logic & Physics-of-Failure Approach to Reliability Prediction. *IEEE Transactions on Reliability*, 44(2):237–242, 1995.

[2] John B. Bowles. Applying Fuzzy Logic to the Design Process. In *Proceedings Annual Reliability and Maintainability Symposium Tutorial Notes*, pages 1–16, 1996.

[3] Mark A. Boyd. What Markov Modeling Can Do for You. In *Proceedings Annual Reliability and Maintainability Symposium Tutorial Notes*, pages 1–25, 1996.

[4] Kai-Yaun Cai. System Failure Engineering and Fuzzy Methodology An Introductory Overview. *Fuzzy Sets and Systems*, 83(2):113–133, 1995.

[5] Wen Chun-Yaun Cai Kai-Yuan and Zhang Ming-Lian. Fuzzy variables as a basis for a theory of fuzzy Reliability in the Possibility ontest. *Fuzzy Sets and Systems*, 42(2):145–172, 1991.

[6] B. A. Carteret. Light Duty Utility Arm System Applications for Tank Waste Remediation. Technical report, Westinghouse Hanford Company, 1994.

[7] Geoff Cunliffe. Light Duty Utility Arm for Undeground Storage Tank Inspection and Characterisation. Technical report, Spar Aerospace Ltd., 1994.

[8] Luis M. de Campos and M. Jorge Bola nos. Characterization and Comparison of Sugeno and Choquet Integrals. *Fuzzy Sets and Systems*, 52(1):61–67, 1992.

[9] W. Denson, G. Chandler, W. Crowell, A. Clark, and P. Jaworski. *Nonelectric Parts Reliability Data. Technical Report NPRD-95.* Reliability Analysis Ceter, Rome, NJ, 1994.

[10] John A. Drakopoulos. Probabilities, Possibilities, and Fuzzy Sets. *Fuzzy Sets and Systems*, 75:1–15, 1995.

[11] Joanne Bechta Dugan and Stacy A. Doyle. New Results in Fault-Tree Analysis. In *Proceedings Annual Reliability and Maintainability Symposium Tutorial Notes*, pages 1–23, 1996.

[12] Boris Gnedenko and Igor Ushakov. *Probabilistic Reliability Engineering.* John Wiley & Sons, 1995.

[13] John D. Healy. Basic Reliability. In *Proceedings Annual Reliability and Maintainability Symposium Tutorial Notes*, pages 1–16, 1996.

[14] Karel Hrbacek and Thomas Jech. *Introduction to Set Theory.* Marcel Dekker, Inc., 1984.

[15] S. Jagannathan and F.L. Lewis. Discrete-Time Adaptive Fuzzy Control of Robotic Systems. In *Proceedings of the 1996 IEEE Conference on Robotics and Automation*, pages 2586–2591, 1996.

[16] Chang Eun Kim. Markov Decision Process: Fuzzy Approach. In *Computers & Industrial Engineering Proceedings of the 16th Annual Conference on Computers and Industrial Engineering*, volume 27, pages 161–165, 1994.

[17] Lawrence M. Leemis. *RELIABILITY Probabilistic Models and Statistical Methods.* Prentice-Hall, 1995.

[18] Roger D. Leitch. *Reliability Analysis for Engineers.* Oxford University Press, 1995.

[19] Martin L. Leuschen, Ian D. Walker, and Joseph R. Cavallaro. Robot Reliability Using Fuzzy Fault Trees and Markov Models. In *SPIE Proceedings on Sensor Fusion and Distributed Robotic Agents*, volume 2905, pages 73–91, 1996.

[20] Sergey V. Gurov Lev V. Utkin. A General Formal Approach for Fuzzy Reliability Analysis in the Possibility Context. *Fuzzy Sets and Systems*, 83:203–213, 1996.

[21] Kenneth G. Manton, Max A. Woodbury, and H. Dennis Tolley. *Statistical Applications Using Fuzzy Sets.* John Wiley & Sons, 1994.

[22] Milan Mareš. *Computation Over Fuzzy Quantities.* CRC Press, 1994.

[23] Magdi Mohamed and Paul Gader. Generalization of Hidden Markov Models Using Fuzzy Integrals. In *Proceedings of the 1994 1st International Joint Conference of NAFIPS/IFIS/NASA*, pages 3–7, 1994.

[24] Takehisa Onisawa. An Application of Fuzzy Concepts to Modelling of Reliability Analysis. *Fuzzy Sets and Systems*, 37(2):267–286, 1990.

[25] Takehisa Onisawa and Janusz Kacprzyk, editors. *Reliability and Safety Analyses under Fuzziness.* Physica-Verlag, 1995.

[26] Takehisa Onisawa and Yasushi Nishiwaki. Fuzzy Human Reliability Analysis on the Chernobyl Accident . *Fuzzy Sets and Systems*, 28(2):115–127, 1988.

[27] Kyung S. Park and Ji S. Kim. Fuzzy Weighted-Checklist with Linguistic Variables. *IEEE Transactions on Reliability*, 39:389–393, 1990.

[28] V. Venkat Raj P.V. Suresh, A.K. Babar. Uncertainty in Fault Tree Analysis: A Fuzzy Approach. *Fuzzy Sets and Systems*, 83(2):135–141, 1996.

[29] H. Schneider and P.M. Frank. Fuzzy Logic Based Threshold Adaption for Fault Detection in Robots. In *Proceedings of the 1994 IEEE Conference on Control Applications*, pages 1127–1132, 1996.

[30] D. Singer. A Fuzzy Set Approach to Fault Tree and Reliability Analysis. *Fuzzy Sets and Systems*, 37(2):267–286, 1990.

[31] SPAR. Light Duty Utility Arm LDUA and Deployment System - Detailed Design Report. Document SPAR-LDUA-R.025. Technical report, Spar Aerospace Ltd., 1994.

[32] M.W. Spong and M. Vidyasagar. *Robot Dynamics and Control.* John Wiley & Sons, 1989.

[33] Joe Sullivan and William H. Woodall. A Comparison of Fuzzy Forcasting and Markov Modeling. *Fuzzy Sets and Systems*, 64(3):279–293, 1994.

[34] H. Tanaka, L.T. Fan, F.S.Lai, and K. Toguchi. Fault Tree Analysis for Fuzzy Probability. *IEEE Transactions on Reliability*, R-32(5)(December):453–457, 1983.

[35] Mihaela Ulieru. From Fault Trees to Fuzzy Relations in Managing Heuristics for Technical Diagnosis. *IEEE International Conference on Systems, Man, and Cybernetics*, 2:392–397, 1983.

[36] Igor A. Ushakov. *Handbook of Reliability Engineering.* John Wiley & Sons, 1994.

[37] Ian D. Walker and Joseph R. Cavallaro. Failure Mode Analyses of the ORNL MLDUA Manipulator and Hose Management System. Technical report, Rice University, 1996.

[38] Ian D. Walker and Joseph R. Cavallaro. The Use of Fault Trees for the Design of Robots for Hazardous Environments. In *Proceedings Annual Reliability and Maintainability Symposium*, pages 229–235, 1996.

[39] Fei-Yue Wang Xiaobo Shi, Paul J. A. Lever. Experimental Robotic Excavation with Fuzzy Logic and Neural Networks. In *Proceedings of the 1996 IEEE Conference on Robotics and Automation*, pages 957–962, 1996.

[40] Yuji Yoshida. Markov Chains With a Transition Possibility Measure and Fuzzy Dynamic Programming. *Fuzzy Sets and Systems*, 66(1):39–57, 1994.

[41] L. Zadeh. Fuzzy Sets. *Information and Control*, 8:338–353, 1965.

[42] Deli Zhang and Zixiao Wang. Fuzzy Integrals of Fuzzy Valued Functions. *Fuzzy Sets and Systems*, 54(1):63–67, 1993.