An Evidence Theoretic Approach to Design of Reliable Low-Cost UAVs.

By: Justin Fortna Murtha

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Aerospace Engineering

Dr. Craig Woolsey, Committee Chair
Dr. Kevin Kochersberger, Committee Member
Dr. James Marchman III, Committee Member
July 28, 2009
Blacksburg, Virginia

Keywords: Dempster-Shafer Theory, Evidence Theory, UAV, Fault Tree Analysis,
Reliability, SPAARO

An Evidence Theoretic Approach to Design of Reliable Low-Cost UAVs.

Justin Fortna Murtha

Abstract

Small unmanned aerial vehicles (SUAVs) are plagued by alarmingly high failure rates. Because these systems are small and built at lower cost than full-scale aircraft, high quality components and redundant systems are often eschewed to keep production costs low. This thesis proposes a process to "design in" reliability in a cost-effective way. Fault Tree Analysis is used to evaluate a system's (un)reliability and Dempster-Shafer Theory (Evidence Theory) is used to deal with imprecise failure data. Three unique sensitivity analyses highlight the most cost-effective improvement for the system by either spending money to research a component and reduce uncertainty, swap a component for a higher quality alternative, or add redundancy to an existing component. A MATLAB® toolbox has been developed to assist in practical design applications. Finally, a case study illustrates the proposed methods by improving the reliability of a new SUAV design: Virginia Tech's SPAARO UAV.

Acknowledgements

There are many people whom I'd like to thank for supporting me in my studies and research, and many more that have helped form me into the person I am today. These people were there for me whenever I needed guidance or motivation and supported me along the way.

First, I'd like to thank my committee members for their constant enthusiasm in my research and related projects. I'd especially like to thank Dr. Craig Woolsey who's been an invaluable advisor to me for undergrad and graduate studies. He's allowed me (among others) the opportunity and freedoms to develop multiple UAV systems and in doing so, I've realized my true passion for UAV design and flight testing.

NAVAIR has had a large part in shaping the scope of my research. Without the guidance of Dr. Steve Cook and Pete Heasley, this research wouldn't be possible. I truly thank them and the Airworthiness Office for the opportunity they have given me over the past two years and I hope this research is valuable to their airworthiness improvement efforts.

I'd like to thank everyone in the NSL lab for their daily support, especially Laszlo Techy who has worked extensively with me on similar projects and Chris Cotting for being our resident airplane genius, and friend.

Additionally, I'd like to thank Dr. William Mason for his valuable guidance and open door policy throughout my college career and Marty Rothwell whose high school engineering classes were my first exposure to aerospace engineering. Among all I've learned at VT, Dr. Mason's airplane design experiences still fascinate me and Marty Rothwell's "sanity check" has proved to be the most valuable step in any calculation.

Finally, I would like to thank my parents for their constant support and for putting up with all of my airplane banter over the years. I'm truly grateful for their open arms that have formed me into who I am today.

Contents

List of Figures

List of Tables

1 Introduction

1.1 The Value of UAVs

Small Unmanned Aerial Vehicles (SUAVs) are already a vital part of the U.S. military arsenal. 2008 was the first year that scheduled unmanned military sorties outnumbered manned missions. Even the most advanced platforms suffer from alarmingly high failure rates [3].

Major benefits of unmanned aircraft are that they can be smaller, can be built at lower cost, and can be more portable than manned aircraft. But the benefits of SUAVs are offset by their current lack of durability and long term reliability. Often redundant systems are sacrificed for cost savings, and short design cycles drive a "build and test" design process instead of high-fidelity modelling and prediction. These cheap, fast-paced design practices account for the current proliferation of Unmanned Aerial Vehicles (UAVs), but in many cases adversely affect a system's reliability. Is there a way to "design for reliability" in these inexpensive systems, where cost *and* reliability are of utmost importance?

1.2 Current Reliability of UAVs

Without complex safety systems, some UAVs suffer from failure rates nearly 100 times greater than manned airplanes [24]. Figure 1 compares the failure rates based on cumulative flight hours for UAVs such as the Pioneer and Shadow, and also for F-16s and U2s for the years 1984-2004.

Pioneer and Shadow are medium sized UAVs between 200 and 500 lbs, while Global Hawk and Predator are nearly full-scale, multi-million dollar airplanes. It is obvious from this survey that lighter weight, cheaper UAVs have much higher failure rates than full sized UAVs or other manned aircraft. SUAVs are omitted from this chart due to their extremely high failure rates and low tenure in service, underscoring the need to increase their reliability.

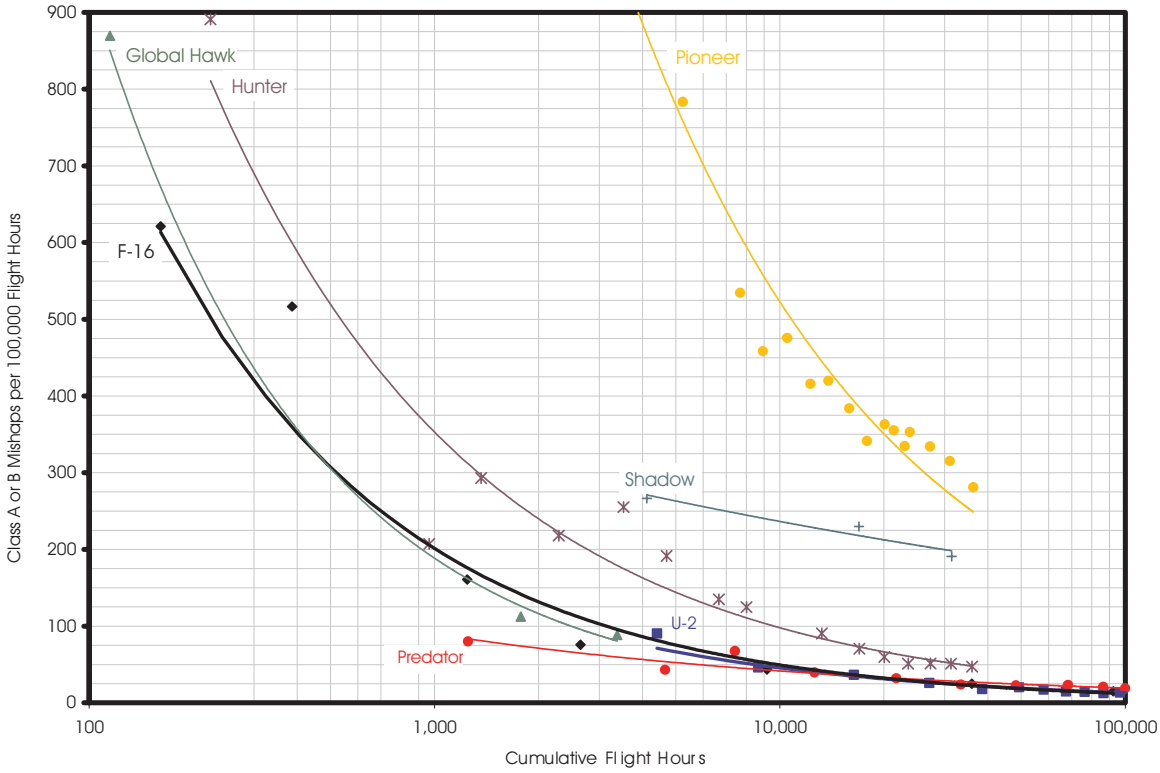


Figure 1: Mishap Rate vs. Cumulative Flight Hours for Various Aircraft [24]

1.3 Practicality of SUAV Reliability Optimization

Developing an extremely reliable yet efficient product requires costly high-fidelity modelling, simulation, and multidisciplinary design optimization (MDO). Industry has proven its competence to design such systems because we do have reliable cars, ships, airliners, etc.. But, this total system optimization is very expensive and often not an option for SUAV engineers who are forced to rely on low-cost, quick turn-around analyses. Therefore, often *no* reliability optimization is performed because it is deemed too expensive or time consuming for SUAV development budgets.

This thesis addresses the conflicting problem of "cost or reliability" by proposing a method to "design in" reliability in a cost-effective way. While extensive design optimization may provide the unquestionable "optimal" design, MDO's expense makes it impractical for most SUAV designs. This thesis suggests an accessible reliability improvement procedure so a SUAV developer can progressively improve the reliability of his current design in a cost-effective (and practical) way.

Fault Tree Analysis is proposed as the basic tool to evaluate a system's (un)reliability, and Dempster-Shafer Theory (Evidence Theory) is incorporated to help deal with imprecise or conflicting failure rate data such as expert opinions. Once a system's initial *probability of failure* (PoF) is evaluated, one can perform sensitivity analyses to highlight which components should be improved or made redundant (or simply studied more carefully) to increase the *system's* reliability in the most cost effective way. In other words, the analyses suggest the most effective sequence of reliability improvements to yield the largest improvement in *system* reliability per dollar spent.

Improving a system's reliability "on the cheap" simply involves iteratively evaluating reliability, improving the most critical component via sensitivity analysis, and recording the actual cost to improve this component. Doing so (either theoretically, at the beginning of a design cycle, or in real time during detailed design), one may develop a "reliability vs. cost" curve that, in principle, asymptotically approaches zero failures at infinite cost. It is up to the

designer to identify the point of diminishing returns based on the budget; at some point the benefits of improving reliability won't be worth the cost to further improve the system. The "elbow" in the "reliability vs. cost" curve is a natural place to stop making improvements, however any point can be chosen as the point of diminishing returns. Section 2.3 discusses this process in further detail.

2 Current Methods to Evaluate Reliability

Evaluating a system's reliability is often done by evaluating it's *un*reliability, or probability of failure in a given period. For complex systems with various stages of missions, "success" becomes hard to define. For a UAV, for example, is success defined as flying for a certain time, reaching a certain altitude, visually inspecting a given target, or is it simply returning safely to base without damage? On the other hand, failure is a much more distinct scenario – if the UAV crashes and is destroyed, everyone will agree that the mission was a failure. For this reason, the proposed methods in this thesis investigate probability of *failure* (PoF) rather than probability of *success*. Further, *failure* will be defined as a total loss of the platform.

2.1 FMEA/FMECA

Failure Mode and Effect Analysis (FMEA) and its sibling Failure Mode and Effect Criticality Analysis (FMECA) are widely used tools for evaluating a system's reliability and failure modes. FMEA is often the first pass at judging a system's reliability because the process takes into account the system as a whole. FMEA uses a *deductive* approach in either a top-down manner (i.e., find the causes of each possible failure), or a bottom up approach (i.e., determine what effect a given failure will have on the system). MIL-STD-1629A is the U.S. Department of Defense's manual for performing FMEA/FMECA [25]. Further, Dermentzoudis has done extensive research on FMEA/FMECA methods to evaluate and improve UAV Reliability [6], and has developed detailed UAV specific fault tree diagrams.

Because of it's deductive emphasis on *system* reliability, FMEA is generally too involved for early design. Early design is a unique time to look at system reliability, however, because specific components have yet to be chosen; many times a generic "component A" serves to represent a component of undetermined specifications. In each design iteration, parts are swapped and the system's architecture is modified leading to an unreasonable number of

FMEA worksheets. For example, if we've filled out a FMEA worksheet for an autopilot's effect on UAV (un)reliability but we propose adding a second, redundant avionics unit to the system, this requires *new* FMEA worksheets for both the original and redundant avionics. That is, failure of either avionics now has *less* of an effect on the system than the original (non-redundant) system. Thus, changing a system architecture often requires previous FMEA worksheets to be modified for every change in system architecture.

In the next section, Fault Tree Analysis is proposed as a more suitable tool for reliability studies in the early design process.

2.2 Fault Tree Analysis

2.2.1 Basic System PoF Calculation

Fault Tree Analysis (FTA) is a mathematically simple Boolean tool for modelling a system's unreliability. It was originally developed in the early 1970's to quantify the safety of the Minuteman Missile System and various nuclear reactors, but has recently been applied to multidisciplinary reliability studies [8].

The conventional analysis, proposed by Vesely, determines the probability of failure (PoF) of a system from a set of components that make up that system. The components are organized in a directed graph structure known as a "tree" with lower cells representing individual components and the top level cell representing a system failure [27]. Boolean AND and OR gates are used to connect components with one another. Figure 2 shows two simple fault trees with an OR and AND gate for a two component system.

For the simple system in Figure 2a, the top event occurs if either component A *or* B fails. Figure 2b is a fault tree with an AND gate, where the top event, T, only occurs if both A *and* B occur; this is an example of a redundant system where failure occurs if both the primary and backup components fail. For an AND gate (Figure 2b), the probability of system failure

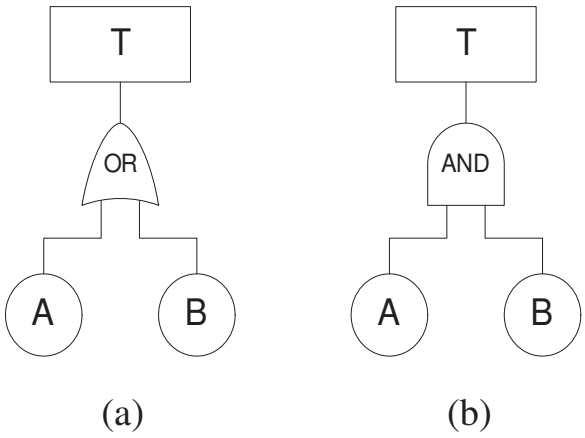


Figure 2: (a) AND gate (b) OR gate

is

$$p_T = p_A * p_B \tag{1}$$

where p_A and p_B are the corresponding failure probabilities for each component, respectively. Likewise, top event failure for an OR gate (Figure 2a) is

$$p_T = p_A + p_B - (p_A * p_B) \tag{2}$$

The third term $(p_A * p_B)$ arises from the chance that *both* components fail at the same time. Without this third term, this chance of simultaneous failure is double counted, artificially increasing the predicted system failure rate. While this term is simple for a two component system, additional OR gates make the expression for p_T more and more complicated. The top event probability for a set of n components connected through OR gates as in Figure 3 is shown in Equations (3)-(4).

$$p_T = p(C_1 \text{ OR } C_2 \text{ OR } C_3 \text{ OR } ...C_n) \tag{3}$$

$$p_T = \sum_{i=1}^{n} p(C_i) - \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} p(C_i * C_j) + \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^{n} p(C_i * C_j * C_k) \ldots + (-1)^n p(C_1 * C_2 * C_3 ... C_n) \tag{4}$$

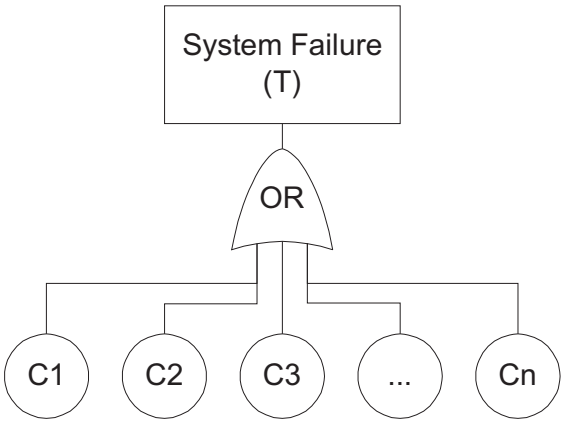


Figure 3: Generic Fault Tree with n Components Through an OR Gate

The *rare-event approximation* is often used to simplify this equation if the top event probability p_T is generated by hand. The approximation simply omits the third term in Equation (2), and all but the first term in Equation (4).

$$p_T = \sum_{i=1}^{n} p(C_i) \tag{5}$$

This approximation is accurate to within about 10% of the true probability when $p(C_i) < 0.1$ – often the case for modern failure data. Further, any errors induced by this approximation are conservative.

The rare event approximation is commonly used for failure analysis fault trees. However, the full formula (Equation (4)) should generally be used for OR gates for the most accurate results if computer assistance is available.

2.2.2 "Building up" The Top Event PoF Equation

The complex algebraic system equation, p_T, (Equation (4)) can be exhaustive to derive. An alternative method exists to numerically solve p_T by "building it up" incrementally rather than forming the full algebraic equation in a single step. We can progressively use the basic

AND and OR gate equations (Equations (1)-(2)) and numeric PoF data to iteratively to build p_T.

To illustrate this, p_T for a simple system is derived in two ways, 1) by deriving the full algebraic p_T equation and 2) "building up" p_T with a series of simplifying substitutions. Figure 4 shows a three-component system connected by an OR gate. The (full) fault tree equation for p_T is represented in Equation (6).

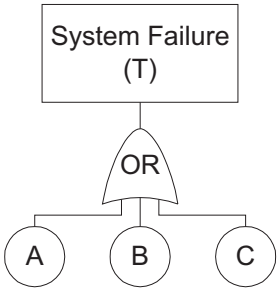


Figure 4: A Simple Three-Component OR Gate Fault Tree

$$p_T = p_A + p_B + p_C - (p_A * p_B) - (p_A * p_C) - (p_B * p_C) + (p_A * p_B * p_C) \tag{6}$$

Suppose we have failure data for components A, B, and C as listed in Table 1.

Component	PoF
A	0.3
B	0.4
C	0.5

Table 1: Failure Data for Components A, B, C

If we've already computed the full system equation (Equation (6)) we can simply substitute the failure probabilities into this equation to get a system PoF:

$$p_T = 0.3 + 0.4 + 0.5 - (0.3 * 0.4) - (0.3 * 0.5) - (0.4 * 0.5) + (0.3 * 0.4 * 0.5) = 0.79. \tag{7}$$

But, this burden to generate the full algebraic equations like Equation (6) isn't required if we build p_T iteratively instead. Suppose we transform the subsystem A OR B into a new, temporary component X, as shown in Figure 5.

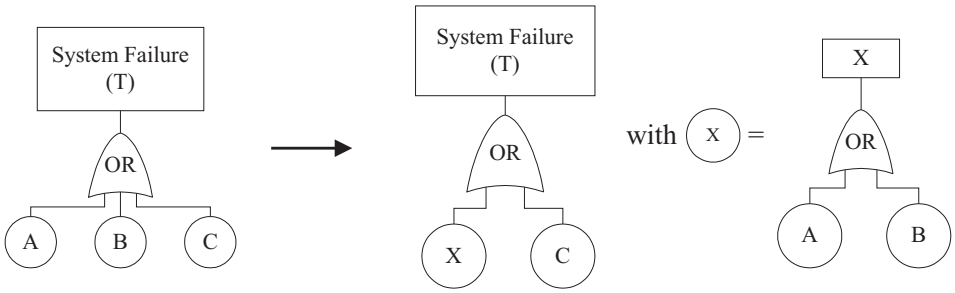


Figure 5: A-B-X Transformation

We first calculate the PoF of this new component X by using the basic equation $p_X = p_A + p_B - (p_A * p_B)$; here, PoF for component X is 0.58. Next, we find p_T, now simplified to $p_T = p_X + p_C - (p_X * p_C) = 0.79$. Note the equivalent result to the basic substitution method, but we only used the basic OR gate equation twice instead of deriving a (relatively) complex algebraic equation for p_T.

In summary, we can always calculate the system PoF incrementally *without* the need to find the complex algebraic equation in the form of Equation (4). This method allows us to use *only* the basic AND and OR gate equations (1)-(2) in a series of steps to compute exact, complex system PoF.

2.3 A Process to Inexpensively Improve Reliability in SUAV Design

Multidisciplinary design optimization (MDO) is a methodical approach for incorporating many competing design objectives into design of a final product. Recent work in the area of MDO for aircraft design includes, for example, work by Sobieszczanski-Sobieski et al. [31], Giunta et al. [12], and Peoples & Willcox [26]. Reliability based design optimization

(RBDO) has been proposed to incorporate failure modes and component (un)reliability into the optimization process [1]. MDO and RBDO are popular among designers with abundant resources, but complete system optimization is often too expensive for SUAV developers with limited budgets.

Rather than focus on "system optimization," a new process is proposed where reliability can be progressively improved through a procedure that results in a more reliable and budget-practical design. An iterative "analyze & improve" procedure results in a sequence of locally cost-effective improvements. Improvements can be made until reaching a point of diminishing returns or until the project's budget is reached. This process consists of the following steps:

1. Generate a fault tree of the current system or design.
2. For each component, assign a description, PoF, cost, and money allotted to improve[1] ($MATI$).
3. Evaluate the system's initial PoF and Average Width, AW (a measure of uncertainty discussed in Section 3.8).
4. Run sensitivity analyses to predict the *theoretical* cost-optimal part to improve (discussed in Chapter 4).
5. Make an improvement by allocating funds to either swap a part with a higher quality alternative, add redundancy, or research the component to reduce uncertainty of the PoF estimate.
6. Record the *actual* cost of this improvement.
7. Plot the *new* PoF from this improved system vs. total money allocated thus far.
8. Iterate steps 1-7 until a stop criterion is satisfied.

[1]MATI represents the additional amount of money available to improve a part. It might be based, for example, on the price increment required to upgrade to a slightly better component.

This process allows an engineer to incrementally improve the system's reliability (and PoF uncertainty) without the need for an MDO routine. This process can be illustrated with a simple example: a battery powered autopilot driving a single servo. The fault tree for this system is shown in Figure 6 where the original system fails if *either* the battery, autopilot, or servo fails.

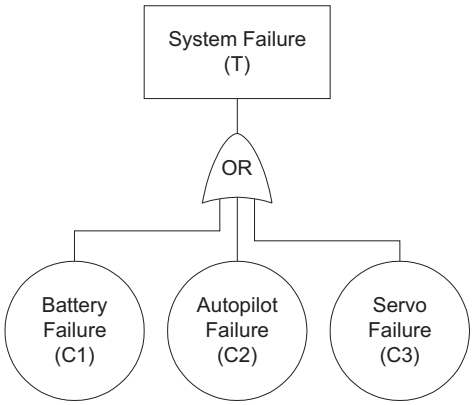


Figure 6: Fault Tree for a Simple Servo Controller

As shown, bottom level components are represented as circles and top level events are rectangles. Generally, a description of each component is written inside the circles as well as an abbreviation for notational simplicity when deriving system equations. For example, the battery in Figure 6 has a description ("Battery Failure") as well as abbreviation (C1) inside its circle.

Assume we have some information for these components, given in Table 2.

Component Description	PoF, λ	Cost, \$	$MATI$, \$
Battery (C1)	0.05	50	10
Autopilot (C2)	0.001	5,000	2,000
Servo (C3)	0.1	30	10

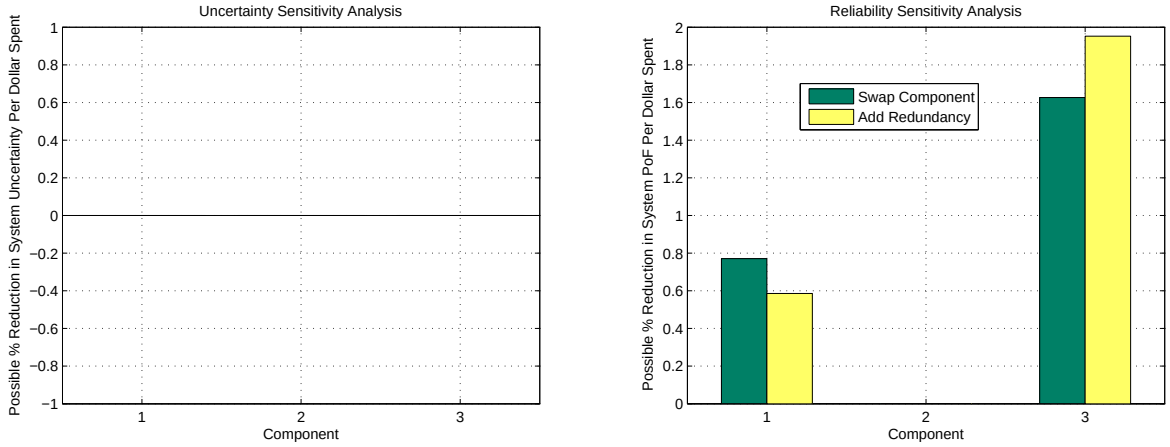
Table 2: Component Data for a Simple Fault Tree

Note that while Cost and $MATI$ are represented here as monetary units, these costs may be defined as a *installation* costs; perhaps adding a redundant part increases the system's weight where simply swapping a component for a higher quality alternative doesn't add any weight. Therefore, Cost and $MATI$ should be defined and interpreted in a way that includes costs due to added structure and complexity.

Using the *rare-event approximation*, the system equation for the fault tree in Figure 6 is simply

$$p_T = p_{C1} + p_{C2} + p_{C3} \tag{8}$$

Using the failure rates from Table 2 and this system equation, the PoF for the top level event (System Failure) is 0.151 with a cost of \$5,080[2]. With an initial PoF, we now examine the sensitivity analysis results in Figure 7.



(a) Sensitivity to Uncertainty

(b) Sensitivity to PoF Reduction via Part Swap or Redundancy

Figure 7: Sensitivity Analysis Results

The uncertainty analysis (Figure 7a) doesn't reveal where we can reduce PoF uncertainty because all PoF input data is deterministic; that is, the component PoFs are precisely

[2]Note, the actual PoF without the *rare-event approximation* for this system is 0.14586, suggesting this approximation is a valid, conservative assumption.

known. The uncertainty sensitivity analysis will be omitted for the remainder of this example because it is always trivial with deterministic input data[3].

From the reliability sensitivity analysis in Figure 7b we see that adding an identical redundant part to Component 3 (the servo) will increase the system reliability the most, per dollar spent. We decide to add an identical redundant servo, so the system Fault Tree architecture changes to that shown in Figure 8.

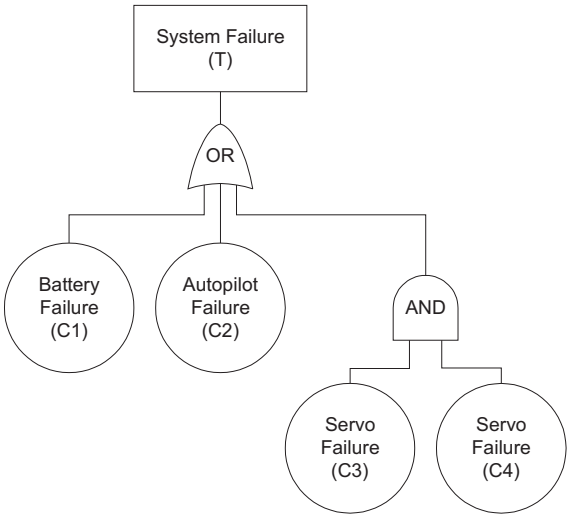


Figure 8: Updated Fault Tree for a Simple Servo Controller (Step 2)

The new system equation corresponding to Figure 8 is:

$$p_T = p_{C1} + p_{C2} + (p_{C3} * p_{C4}) \tag{9}$$

We've added another servo that costs \$30, so our system cost goes up to \$5,110. The PoF of this new system is 0.061.

To further increase the reliability of this system, the sensitivity analysis is run again and results are shown in Figure 9.

Now, the battery (C1) is the next most critical component. The low improvement cost ($MATI$, Table 2) suggests there are many battery alternatives available for relatively small

[3]Section 3 deals with imprecise input data.

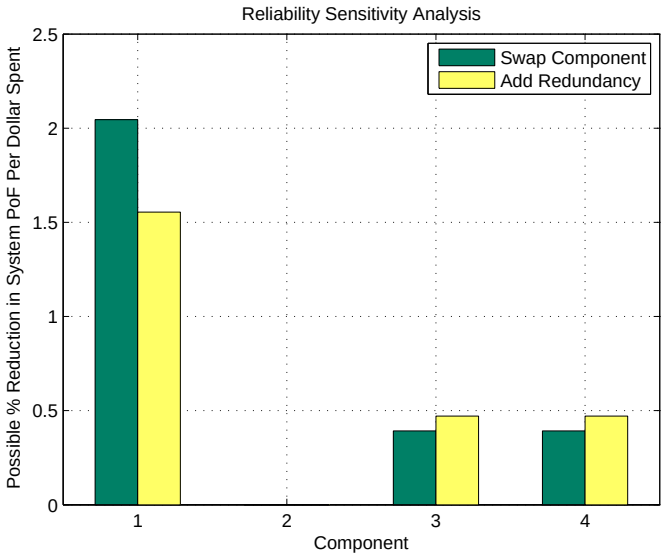


Figure 9: Reliability Sensitivity Analysis (Step 2)

price increases. The sensitivity analysis shows that given the relatively high cost of a battery, it's more cost effective to shop for a higher quality (but slightly more expensive) battery than to add an identical redundant battery.

Suppose we've found a better (but more expensive) battery that costs \$65 rather than \$50 and its PoF is 0.03 instead of the original 0.05. The system Fault Tree doesn't change, but the updated failure data are shown in Table 3.

Component Description	PoF, λ	Cost, \$	$MATI$, \$
Battery (C1)	0.03	65	20
Autopilot (C2)	0.001	5,000	2,000
Servo (C3)	0.1	30	10

Table 3: Updated Component Data for a Simple Fault Tree

The system is re-analyzed and yields a new PoF of 0.041 with a system cost of \$5,125. A sensitivity analysis is run again to show the battery is still the most critical component, but

now adding a redundant part is the best choice.

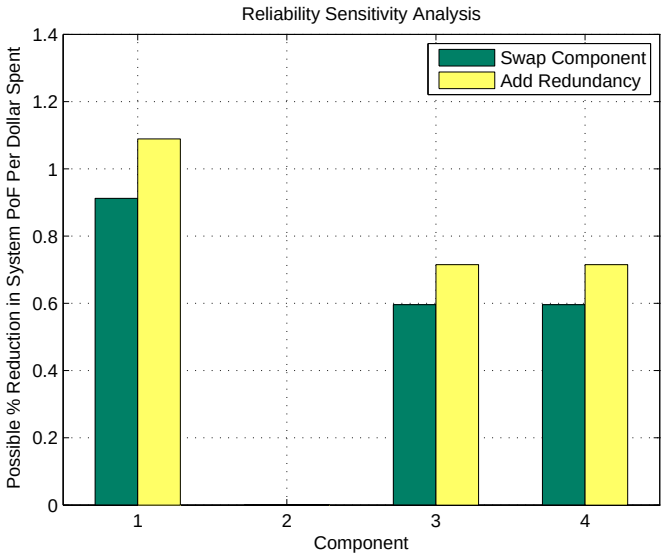


Figure 10: Reliability Sensitivity Analysis (Step 3)

If we add a redundant battery, the system Fault Tree changes to that shown in Figure 11 and the resulting system equation is given in Equation (10).

$$p_T = (p_{C1} * p_{C2}) + p_{C3} + (p_{C4} * p_{C5}) \tag{10}$$

The failure rate is 0.02 with a total system cost of $5,190. Table 4 lists all the improvement steps performed thus far.

Iteration	PoF	Total System Cost, $	Improvement
0	0.151	5,080	–
1	0.061	5,110	Add a redundant servo
2	0.041	5,125	Swap C1 for a better battery
3	0.020	5,190	Add a redundant battery–

Table 4: List of Improvements for a Simple Fault Tree System

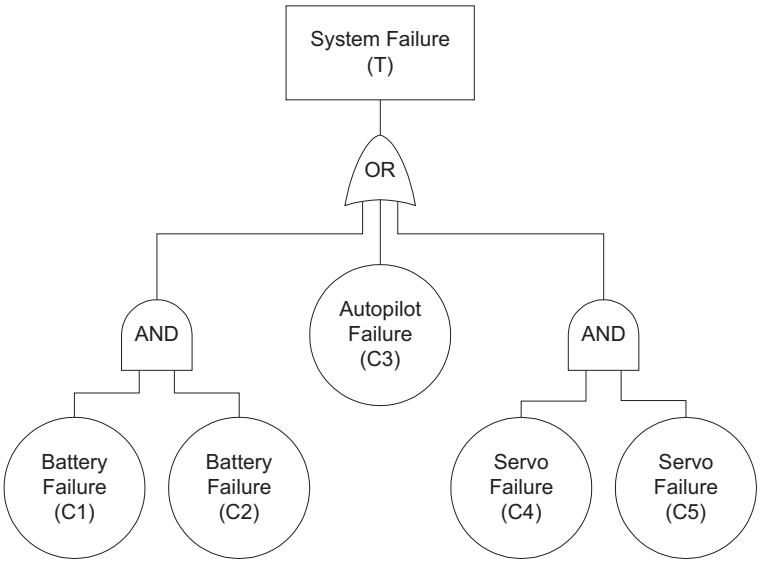


Figure 11: Updated Fault Tree for a Simple Servo Controller (Step 3)

To visualize the effectiveness of these improvements, the system PoF vs. cost is plotted in Figure 12.

We could keep improving and adding redundant components to the system if the project budget (and weight budget) has not been surpassed; however, it could be argued that we should stop after the 2nd improvement and should *not* add a redundant battery. This improvement looks to be *after* the point of diminishing returns on Figure 12.

This process can be used late in the design process where *actual* costs are used or during preliminary design stages where *estimated* costs could be used. Such a method allows reliability to be "designed in" to the SUAV before production begins when improvements are most inexpensive and efficient.

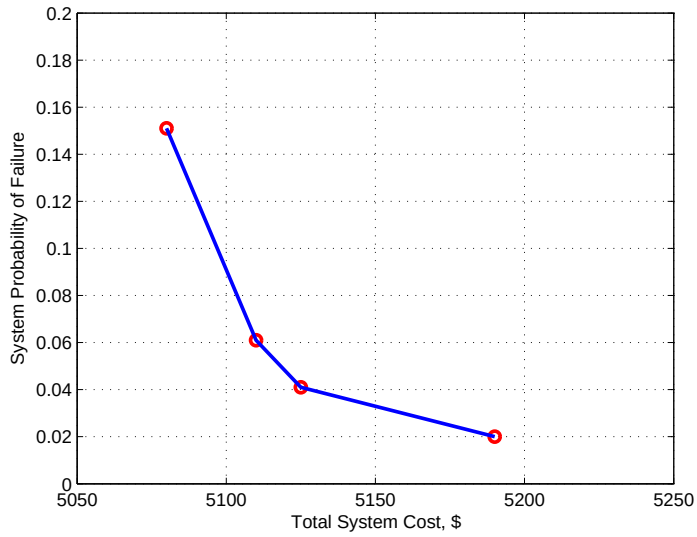


Figure 12: *PoF* vs. System Cost for a Simple Fault Tree System

3 Methods to Utilize Imprecise & Estimated Failure Data

3.1 Epistemic & Aleatory Uncertainty

It is often difficult to obtain accurate failure data for "commercial-off-the-shelf" (COTS) parts that are frequently used in SUAV designs. Schneidewind [28], Kohl [17], and Dumas [7] have researched COTS products' suitability in recent military defense applications. Private companies are developing engines, servos, control systems, structures, etc. that are invaluable for the small UAV designer, but rarely do they come with failure statistics. The UAV designer is left to determine a component's suitability based on missing or incomplete failure statistics – often he is left simply surveying expert users to get estimates for component failure rates. With incomplete failure data, a designer can choose to invest in accelerated life-cycle testing, but testing is expensive and often not worth the investment for smaller, less critical parts.

Uncertainty due to a lack of information is "epistemic uncertainty." Formally, it is uncer-

tainty that is "due to a lack of knowledge of quantities or processes of the system or the environment" [23]. This uncertainty is *reducible*. A component's true probability of failure is often unknown, but this epistemic uncertainty can be reduced with better (more accurate) failure data. Accelerated life-cycle testing is one way epistemic uncertainty can be reduced.

In addition to epistemic uncertainty, aleatory uncertainty also arises in any stochastic system. This *irreducible* or *stochastic* uncertainty is present because any given component will fail at a slightly different time than an identical sibling.

To illustrate the difference between the two types of uncertainty, imagine predicting the period of pregnancy for a human and for some new species "Elmo." Centuries of statistical data have shown that the average female human gives birth nine months from conception, yet any given baby is born after a slightly different period ($\pm$ a few days). While the epistemic uncertainty is low because of the overwhelming data suggesting nine months is an accurate average, there still exists aleatory uncertainty. We can't predict the *exact* pregnancy period for a given person.

On the other hand, we may know nothing of the new Elmo species making prediction very difficult. Here, we have a similar aleatory uncertainty as the human prediction, but we also have epistemic uncertainty – we don't know the typical Elmo gestation period. One could guess, based on the creature's size for example, and a biologist's guess might carry more weight that a non-expert's. But, without detailed statistical analysis, it would still be just a guess.

Fault tree analysis is well suited for aleatory uncertainty because inputs and results are *probabilities* that, by definition, allow aleatory uncertainty. However, Fault Tree Analysis is not inherently well suited for epistemic uncertainty because inputs (PoF) must be deterministic numbers to propagate through to the system level (deterministic failure rates such as p_A and p_B in Equation (1)). Fault Tree Analysis can be augmented by using methods that deal with epistemic uncertainty, e.g. Monte Carlo Simulation or Evidence Theory.

3.2 Monte Carlo Simulation

To deal with both aleatory and epistemic uncertainties, Monte Carlo Simulation is widely used in reliability studies. Metropolis et al. presented foundational research on Monte Carlo simulation in 1949 [21], and the recent boom in computer technology has made the method quite popular [14]. Variable system parameters can be propagated through a system model (Fault Tree) n times to find statistical data about the outcomes. In this process, a sample is taken from each component's PoF distribution at each iteration and is used as the deterministic value for that component in the system equation. Information about the system is then inferred from the final distribution and statistical outcomes from the n trials.

For example, the fault tree from Figure 13, "Pitch Control System," could be evaluated using a Monte Carlo technique. To find failure rates for each component (P1-P8, E1, & E2), a UAV pilot with extensive radio control (r/c) experience was surveyed. Table 5 shows his tabulated failure probability estimates.

Component	MTBF Range (hours)	PoF Range
P1	(400,600)	(0.00167,0.00250)
P2	(9000,10000)	(0.00010,0.00011)
P3	(1000,2000)	(0.00050,0.00100)
P4	(1000,1200)	(0.00083,0.00100)
P5	(400,600)	(0.00167,0.00250)
P6	(400,600)	(0.00167,0.00250)
P7	(200,400)	(0.00250,0.00500)
P8	(200,400)	(0.00250,0.00500)
E1	(150,250)	(0.00400,0.00667)
E2	(100,200)	(0.00500,0.01000)

Table 5: Rascal Failure Data Estimates

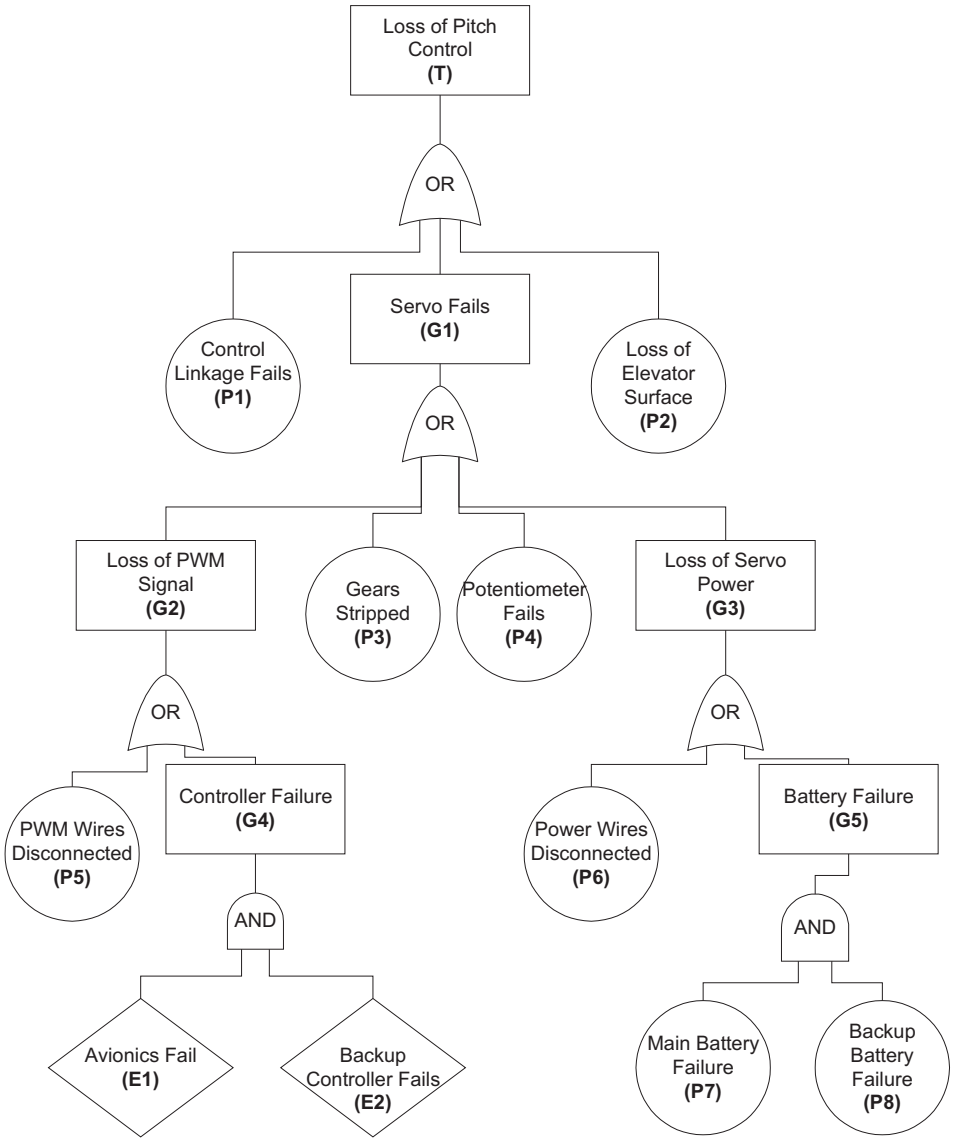


Figure 13: Pitch Control Fault Tree (Rascal UAV at Virginia Tech)

Because no continuous distributions were given, Monte Carlo simulation requires that we assume some distribution for the range. Here, a uniform distribution is assumed over the PoF range for each component. The *principle of insufficient reason*, originally proposed by Bernoulli [13] and later Keynes [16], is often used as a basic assumption in probability theory: with no additional evidence, assume uniform probability through the range. The top event equation for this fault tree (Equation (11)) is solved 10,000 times and the resulting distribution is shown in Figure 14.

$$p(T) = p(P1)+p(P2)+p(P3)+p(P4)+p(P5)+p(P6)+[p(P7)*p(P8)]+[p(E1)*p(E2)] \quad (11)$$

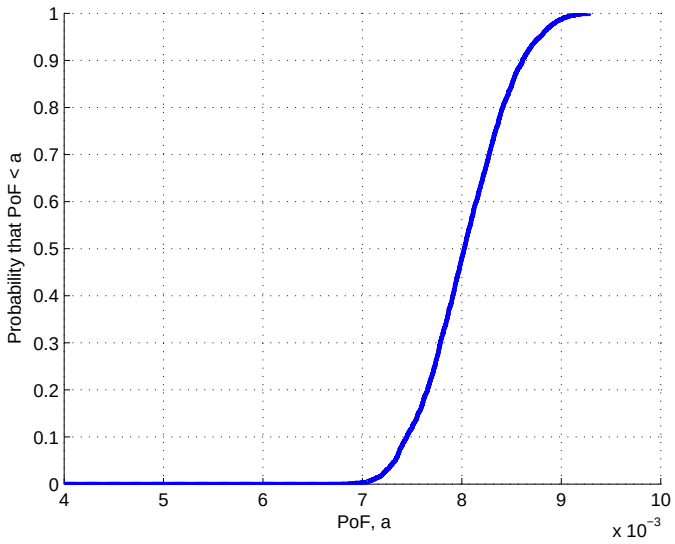


Figure 14: CDF Plot for Pitch Control PoF – Monte Carlo Method

Reliability simulations are commonly displayed as cumulative distribution functions (CDF) rather than probability density functions (PDF) because often we're concerned with the *maximum* predicted failure rate and how much evidence supports this maximum value (to a given confidence). With this CDF plot, one can choose a confidence level (90%, 95%, 99%, etc.) to evaluate the maximum PoF, assuming a given confidence level. On Figure 14 above,

the Monte Carlo simulation shows the PoF of this pitch control system is 8.787e-2 at the 95% confidence level.

In summary, the Monte Carlo method can deal with non-deterministic data by sampling from a distribution, calculating the system solution, and reports it as a deterministic result (for each iteration). Thus, information about the shape of the epistemic input variable (i.e., information about the uncertainty in PoF data) is lost when sampling and reporting the final result.

3.3 Dempster-Shafer Theory (Evidence Theory)

An alternative method of dealing with epistemic uncertainty is Dempster-Shafer Theory (also called Evidence Theory). Arthur P. Dempster [5] and Glenn Shafer [30] wrote the founding literature on the theory in the late 1960's as an alternative to traditional Bayesian or probabilistic theories.

Dempster-Shafer Theory (DST) was originally used with sensor fusion in artificial intelligence. While developed in the late 1960s, the theory has only recently been applied to reliability and dependability models. More recently in 2002, researchers studied use of DST in uncertainty studies related to sensor fusion [32]. Their case studies solved benchmark problems for propagating epistemic and aleatory uncertainty based on DST. It has yet to be fully integrated into the early design process or fluently married with an aleatory reliability model such as FTA.

In DST, the probabilistic PDF (the typical distribution for an unknown variable) is replaced with sets of bounds or ranges – also called Dempster-Shafer Variables (DSVs). The epistemic uncertainty representation is derived from field data or expert estimates that yield an upper and lower bound on probabilities, rather than a traditional probability curve with some assumed parameters (such as mean and variance). Thus, a single curve with specific parameters is not required. A joint DSV can be "built" from the aggregation of all relevant

evidence, without the need for over-constraining assumptions (such as a Gaussian PDF).

While DST supports continuous functions, it is easier to understand the logic and compare it with conventional probabilistic theory by examining discrete cases. Moreover, we will only consider real scalar random variables here.

Conventional probabilistic modelling represents the uncertainty of a variable x using a random variable X. (Think of X as an estimate of the unknown variable x.) The set $A \subset \mathbb{R}$ is composed of all possible values of X. Mass m can be associated with each possible value of $X \in A$, expressing the probability $P(X = x)$. The following rules must be satisfied:

$$m : X \to [0, 1] \tag{12}$$

$$\sum_{X \in A} m(X) = 1 \tag{13}$$

Equation (12) states that the belief (mass) of each value X in the set A must be between 0 and 1. Equation (13) states that all beliefs (the total mass) of A must sum to 1, as a foundation of basic probability theory. In other words, the sum of our belief in all possible values of X must be 100%; x is certain to take *some* specific value $X \in A$.

By extension, the probability that x lies in the interval $[a, b]$ is then:

$$P(x \in [a, b]) = \sum_{X \in [a,b]} m(X) \tag{14}$$

In contrast, Dempster-Shafer theory allows more flexibility among the possible values of X. One may consider situations where X does not take a specific value in A, but some subset of possible values. The new set containing subsets of X combinations is defined as the "power set" of A and is denoted as $\mathcal{P}(A)$. For example, if set $A = \{x, z\}$, then $\mathcal{P}(A) = \{\{\}, \{x\}, \{z\}, \{x, z\}\}$ where $\{\}$ is an empty set (also denoted as $\emptyset$). The formalized rules for Dempster-Shafer structures are as follows:

$$m(\emptyset) = 0 \tag{15}$$

$$\sum_{X \subseteq \mathcal{P}(A)} m(X) = 1 \tag{16}$$

Equation (15) states the mass of the null-space is zero – that is, no mass should be assigned to events that cannot occur (for example, $PoF < 0$ or $PoF > 100\%$). Equation (16) states that the total mass of the power set of A must be 1 – the sum of all of our mass values must be 100%. Further, we make no assumption about the distribution between any given X values. If our X values were given as bounding values then we should not assume any distribution between these values – doing so would introduce unnecessary assumptions like Bernoulli's principle of insufficient reason.

Instead of assigning mass to each value of X (as we do for conventional probabilistic analysis), we assign mass to various combinations (subsets) of X. We have some confidence (mass) that the true value x lies within the bounds of a given subset, but we don't assign mass to any individual value between these bounds.

On the simplest level, consider a Boolean statement where something is either true or false. According to probabilistic theory, our set appears as $\{p_{\text{true}}, p_{\text{false}}\}$ and $p_{\text{true}} + p_{\text{false}} = 1$. In DST, all possible subsets have masses assigned to them, so the new set is thus $\{p_{\text{true}}, p_{\text{false}}, p_{(\text{true OR false})}\}$ where $p_{\text{true}} + p_{\text{false}} + p_{(\text{true OR false})} = 1$. The added freedom from the third term allows for epistemic uncertainty.

In the context of uncertain failure rates, we may have many overlapping or disjoint ranges from various sources. Each range has a maximum and minimum value, and DST allows us to work with these ranges without assuming a distribution throughout the range – the entire analysis can be performed solely with the endpoints of each range, rather than fictional, assumed points in between.

3.4 Data Input for DST

Given the basic differences between probabilistic theory and DST, how can we organize PoF estimates and assign masses (i.e., levels of "trust") to each estimate? (Perhaps we trust one source more than another.) The simplest data to input is a range $[a, b]$ where we have some

confidence the true value x (i.e., the PoF) lies within this range. Such data are called "focal elements" and can be organized together to form a set of Dempster-Shafer Variables (DSVs).

A focal element is a set of upper and lower bounds with non-zero mass $m([a, b]) > 0$. Expert data will yield an upper and lower bound for x (PoF), and these bounds form the "best" and "worst" case scenarios, respectively, in terms of PoF. Mass can be interpreted as our subjective trust in any given set of bounds (or "best-worst case scenario"). Inside the interval $[a, b]$ no distribution is assumed. Given this set, we define a focal element by:

$$\text{Focal Element} = \{Plausibility, Belief, Mass\} \tag{17}$$

where $Plausibility$ is the lower value of the range (a), and $Belief$ is the higher value of the range (b), and $Mass$ is assigned corresponding to how much we trust the estimates of $Plausibility$ and $Belief$. Recall the sum of all masses must be 1 (from Equation (16)).

We also define a Dempster-Shafer variable (DSV) as being a collection of at least one focal elements:

$$\text{DSV} = \left\{ \begin{array}{l} \text{Focal Element 1} \\ \text{Focal Element 2} \\ \text{Focal Element 3} \end{array} \right\} = \left\{ \begin{array}{l} \{\text{Plausibility1, Belief1, Mass1}\} \\ \{\text{Plausibility2, Belief2, Mass2}\} \\ \{\text{Plausibility3, Belief3, Mass3}\} \end{array} \right\} \tag{18}$$

We can plot focal elements by drawing rectangles whose height represents mass. Figure 15 shows five various focal elements.

From Figure 15, we can see that a single value of x could be included in several focal elements, whereas with probabilistic theory, a specific value of x has only a single mass assigned to it. This key concept alleviates the constraint that something either happens, or it doesn't. With basic probability theory, if we are 40% confident that a statement is true, then we are 60% confident that it is not true. However with DST, if we are 40% confident in a hypothesis, the other 60% remains uncommitted to other possibilities.

Recall that masses are assigned to sets of bounds without distributing the mass on any individual values inside the interval. Therefore, it is not possible to find the exact mass that

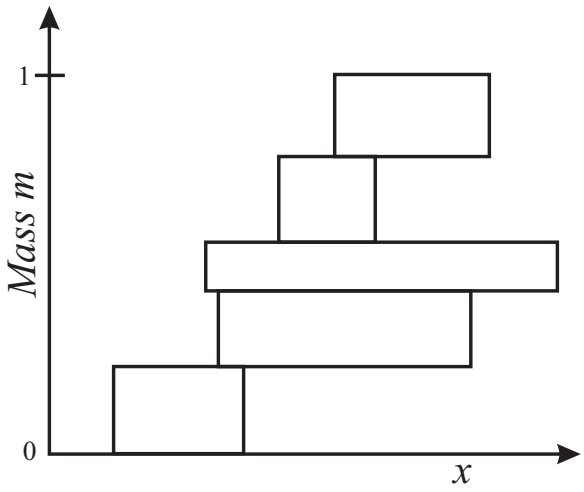


Figure 15: Graphical Representation of Five Focal Elements

suggests $X = x$ or the mass suggesting $x \in [a, b]$. However, upper and lower bounds on these masses may be calculated.

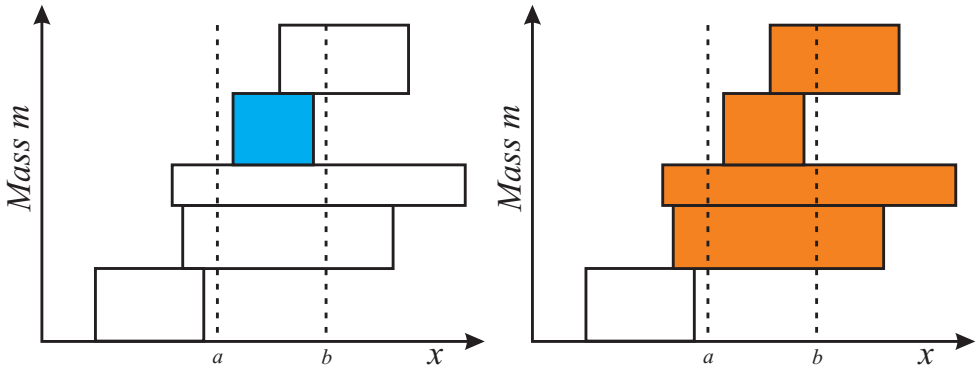
Associated with each DSV (evidential set) are two functions – *Belief* (Bel) and *Plausibility* (Pl).

$$Bel(X \in [a, b]) = \sum_{X \subseteq [a,b]} m(X) \tag{19}$$

$$Pl(X \in [a, b]) = \sum_{X \subseteq \mathcal{P}(A): X \cap [a,b] \neq \emptyset} m(X) \tag{20}$$

The *Belief* function represents the amount of evidence supporting claims that x is completely contained within $[a, b]$. The *Plausibility* function represents the amount of evidence supporting claims that x is contained in the interval spanned by the focal elements that intersect $[a, b]$. Figure 16 shows the focal elements contributing to the *Belief* and *Plausibility* functions.

These functions give us upper and lower bounds on the mass suggesting x (PoF) is within some given range $[a, b]$.



(a) Belief (b) Plausibility

Figure 16: Graphical Representation of Belief & Plausibility

3.5 Basic Probability Assignment (BPA) Plots

A common way to visualize the *Belief* and *Plausibility* mass values from a set of focal elements (a DSV) is to plot the *Belief* and *Plausibility* functions in analogy to a cumulative distribution function (CDF) [19]. This plot is called a Basic Probability Assignment plot (BPA plot). Transforming the *Plausibility* and *Belief* functions into a CDF eliminates the need to keep track of both upper and lower bounds for both *Belief* and *Plausibility* functions. (We now find $mass(x < a)$ instead of $mass(x \in [a, b])$ so we don't need to keep track of a *and* b, but now just some new a.) Additionally, transforming the functions to CDFs results in functions with monotonic "growth".

Further, CDFs are commonly used in probabilistic settings during the decision making process. This transformation remains valid in reliability modelling because we care mostly about the *maximum PoF* we are likely to see. *Plausibility* becomes the maximum *PoF* predicted by the lower bound evidence, and *Belief* becomes the maximum *PoF* from the most conservative (upper bound) evidence.

An example of a simple BPA plot is illustrated for a single focal element in Figure 17. We need to estimate the failure rate for an untested motor. Suppose we guess that the *PoF* lies in $\{0.01, 0.02\}$. We have only a single range, so 100% of our trust is associated with that

evidence ($mass = 1$). The DSV is therefore {0.01,0.02,1}.

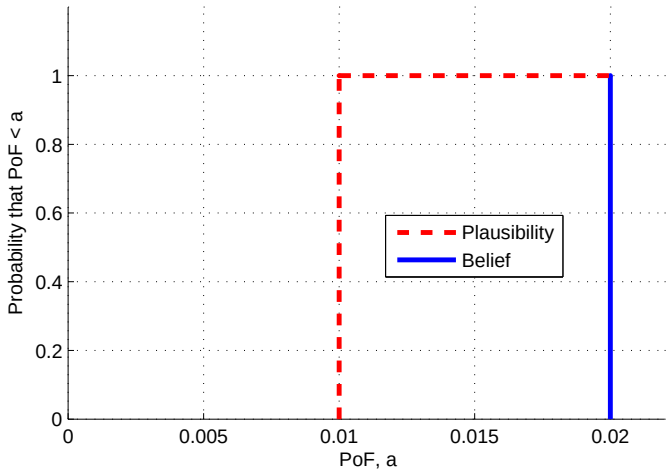


Figure 17: BPA for a Single Focal Element

The horizontal axis is PoF, and any given value is denoted with the variable a. The vertical axis shows $P(PoF < a)$, or equivalently the mass (amount of evidence) suggesting $PoF < a$. Thus, according to available evidence, it is *plausible* that the failure rate is at least 0.01 and it is *certain* that the failure rate is no greater than 0.02. We see from Figure 17 that the BPA plot is simply the visual representation of our single focal element because only a single element (set of bounds) was given. (The height of the focal element is 1 and the end points are shown accordingly.)

3.6 Combination of Evidence

If we have little evidence about the failure rate of a component, we might feel more comfortable getting a second or third opinion. Doing so adds focal elements, so we must combine and normalize our confidence in these elements to obey the basic rules for DST (all evidence sums to 100%). Sentz et al. discuss various ways of combining evidence [29]. Dempster's Rule was originally formulated to combine various sets of evidence, but has been criticized for inadequately handling disjoint or conflicting sets of data. The method of weighted mixing

is commonly used to combine sets of any data. This rule simply normalizes mass of all focal elements:

$$m_i = m_i * \left(\sum_{i=1}^{n} m_i \right)^{-1} \tag{21}$$

where m_i is any given focal element's mass and n is the number of focal elements in the DSV.

To generate a BPA plot with two or more focal elements, we extend the previous example for two focal elements. Suppose we have the original estimate that $x \in \{0.01, 0.02\}$ but we only have 30% confidence in that range. Another expert has come forward who believes the PoF is bounded by the set {0.015,0.02}. We have 70% confidence in the second estimate. Therefore, the focal elements become $\{0.01, 0.02, 0.3\}$ and $\{0.015, 0.02, 0.7\}$ and the DSV for this data is:

$$\text{DSV} = \left\{ \begin{array}{ccc} 0.01 & 0.02 & 0.3 \\ 0.015 & 0.02 & 0.7 \end{array} \right\} \tag{22}$$

Figure 18 plots this DSV.

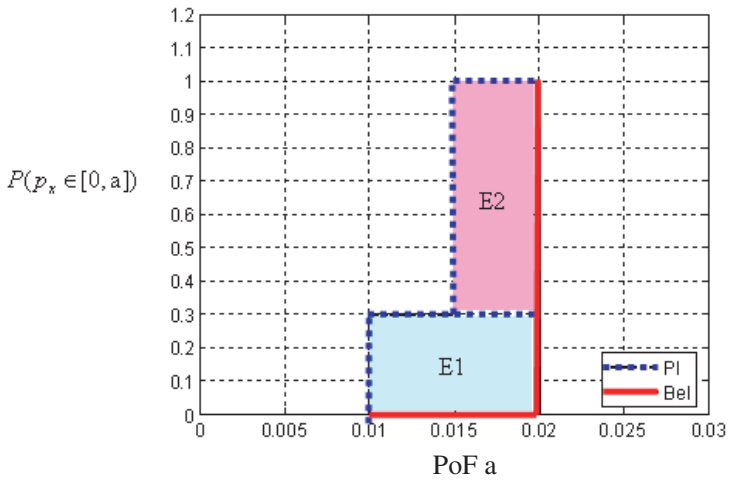


Figure 18: BPA for Two Focal Elements

From this aggregated BPA plot, we can conclude that 100% of our evidence says the worst-case failure rate is less than 0.02, and best-case failure rate is 0.015. Further, 30% of the

evidence says it could be as low as 0.01.

It is fairly easy to plot simple focal elements, but the true value of BPA plots is more easily realized with a more complex example: an AND gate.

The system equation for two components connected through an AND gate is shown in Equation (23):

$$p_T = p_A * p_A \tag{23}$$

Assume we have two independent servos A and B and we ask two experts for a range of "hours to first failure" for each component. Experts 1 and 2 give us ranges of $[200, 400]$ and $[300, 500]$ for servo A, respectively. Probability of failure (per hour) can be simply calculated from the inverse of time to first failure[4]. For example, 200 hours suggests 0.005 failures per hour and 400 hours suggests 0.0025 failures per hour. We have equal confidence in each expert's opinion, so we assign a subjective weight of 50% to each expert[5]. Thus, the DSV for servo A becomes:

$$\text{Servo}A : \left\{ \begin{array}{ccc} 0.0025 & 0.005 & 0.5 \\ 0.002 & 0.0033 & 0.5 \end{array} \right\} \tag{24}$$

Likewise, we survey the experts for their opinions on servo B. This time, Expert 2 has a career of experience with servo B, so we will assign him a subjective weight of 80% compared to Expert 1's 20%. Experts 1 and 2 give ranges of $[300, 600]$ and $[250, 300]$, respectively. The DSV for servo B is then:

$$\text{Servo}B : \left\{ \begin{array}{ccc} 0.00167 & 0.0033 & 0.2 \\ 0.0033 & 0.004 & 0.8 \end{array} \right\} \tag{25}$$

Recall the system equation is $p_T = p_A * p_B$, or $T = A * B$ for notational simplicity. We represent A's *first* DSV entry *lower bound* (component A's lower PoF value from Expert 1) as A_{L1}. To find T, we simply substitute all possible combinations of A and B into the system equation; we have two expert opinions (two focal elements) for each component A

[4]Also referred to as Mean Time Before Failure (MTBF)

[5]Recall that the weight of all sources must sum to 1

and B, so we have the following set of equations:

$$T_{\text{L1}} = A_{\text{L1}} * B_{\text{L1}} \tag{26}$$

$$T_{\text{L2}} = A_{\text{L1}} * B_{\text{L2}} \tag{27}$$

$$T_{\text{L3}} = A_{\text{L2}} * B_{\text{L1}} \tag{28}$$

$$T_{\text{L4}} = A_{\text{L2}} * B_{\text{L2}} \tag{29}$$

Such equations generate the four possible lower bound PoF for the system DSV, T. Note that the system equation is $T = A * B$ because the components are related by an AND gate. If they were related with an OR gate, the system equation (and thus Equations (26)-(29)) would have the form $T = A + B - (A * B)$ rather than $T = A * B$.

To find the corresponding upper bounds for T's DSV, we find the following:

$$T_{\text{U1}} = A_{\text{U1}} * B_{\text{U1}} \tag{30}$$

$$T_{\text{U2}} = A_{\text{U1}} * B_{\text{U2}} \tag{31}$$

$$T_{\text{U3}} = A_{\text{U2}} * B_{\text{U1}} \tag{32}$$

$$T_{\text{U4}} = A_{\text{U2}} * B_{\text{U2}} \tag{33}$$

The mass is also aggregated for these entries. To calculate the corresponding mass, we *always* multiply mass together for both AND *and* OR gates:

$$T_{\text{Mass1}} = A_{\text{Mass1}} * B_{\text{Mass1}} \tag{34}$$

$$T_{\text{Mass2}} = A_{\text{Mass1}} * B_{\text{Mass2}} \tag{35}$$

$$T_{\text{Mass3}} = A_{\text{Mass2}} * B_{\text{Mass1}} \tag{36}$$

$$T_{\text{Mass4}} = A_{\text{Mass2}} * B_{\text{Mass2}} \tag{37}$$

In summary, we have our top level DSV as shown below:

$$T = \left\{ \begin{array}{lll} (A_{L1} * B_{L1}) & (A_{U1} * B_{U1}) & (A_{\text{1Mass}} * B_{\text{1Mass}}) \\ (A_{L1} * B_{L2}) & (A_{U1} * B_{U2}) & (A_{\text{1Mass}} * B_{\text{2Mass}}) \\ (A_{L2} * B_{L1}) & (A_{U2} * B_{U1}) & (A_{\text{2Mass}} * B_{\text{1Mass}}) \\ (A_{L2} * B_{L2}) & (A_{U2} * B_{U2}) & (A_{\text{2Mass}} * B_{\text{2Mass}}) \end{array} \right\} = \left\{ \begin{array}{ccc} 0.000004175 & 0.0000165 & 0.1 \\ 0.00000825 & 0.00002 & 0.4 \\ 0.00000334 & 0.00001089 & 0.1 \\ 0.0000066 & 0.0000132 & 0.4 \end{array} \right\}$$

This system's BPA is shown in Figure 19.

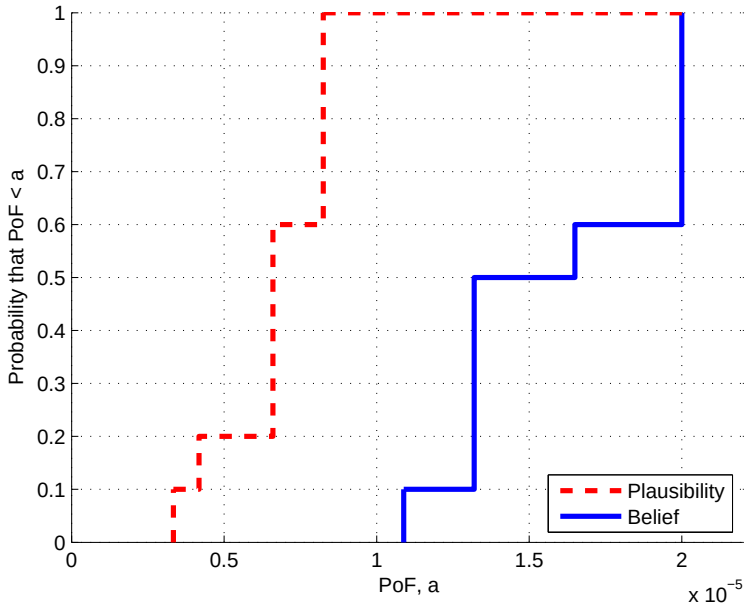


Figure 19: AND Gate System BPA

An exact solution for focal element extrema can be found. In other words, system bounds can be generated for each possible combination of source data, and we could simply "connect the dots" on the BPA plot. When the amount of evidence or components gets very large, the number of resulting system focal elements grows tremendously. The number of focal elements follows n^m where n is the number of sources for each component, and m is the number of components. This equation only holds if n is equal for each component. It must be modified if some sources have more evidential statements than others. For example, a 10 component fault tree with 2 sources of evidence for each component will have 1024 focal elements in the system DSV. Further, simply adding another source of evidence to each of the 10 components raises the number of elements to 59,049. For this reason, computation time can be improved with a reduction in the number of sources.

If a source chooses to give failure probabilities in the form of some distribution with uncertain parameters, a focal element can represent the range of parameters to build the curve. Finally,

the curve can be discretized into n individual focal elements. Limbourg has developed an Imprecise Probability Toolbox (IPPToolbox) for MATLAB® that follows a modified procedure for approximating the system DSV [18]. Instead of calculating every possible focal element for the system (as was presented earlier in this section), Limbourg proposes augmenting the system DSV calculations with Monte Carlo simulation. For n trails, the simulation chooses a single focal element (a single PoF range) to assign to the component, based on a bias from the assigned mass values. To illustrate this on the previously mentioned example with two servos, the simulation calculates SystemDSV=ServoA*ServoB n times. ServoA and ServoB are focal elements picked randomly (although biased from mass values) from either source. To generate the full system BPA plot, all iterations are aggregated and plotted together.

While both techniques are similar (direct system DSV computation vs. an augmented Monte Carlo approach), they are suitable for different applications. For large, complex systems Limbourg's IPPToolbox is useful to efficiently create an "approximate" BPA plot because Monte Carlo simulation will only perform the prescribed number of trials (and thus calculations). To increase accuracy, a large number (> 10000) must be performed. However, the direct calculation method is useful for smaller systems that are more likely to be used in an early design stage. For simple systems, this direct calculation method yields an exact system BPA plot with fewer calculations necessary for accuracy.

3.7 System BPA Plots

Often, systems must be categorized in standardized levels of reliability and safety. A system must be within a certain probability of failure to be categorized in certain Safety Integrity Level (SIL) [19]. For illustration purposes, we define sample SIL levels from 1 (extremely good) to 4 (poor). These levels can determine where the system is allowed to operate; for example, perhaps only UAVs with SIL 1 & 2 can operate in the National Airspace, while SIL 1, 2, & 3 are allowed in deserted areas where the risk of collateral damage is minimal.

Limbourg *et al* discuss these reliability requirements and have provided an extensive example of complex system BPA plots in [19]. If the design calls for the system to be rated to at least SIL 2, we can add reliability constraints to the BPA model to graphically verify if our system is within that range.

An example BPA plot is shown in Figure 20 for the Rascal UAVs in the Nonlinear Systems Lab at Virginia Tech. Vertical dash-dot lines were added that represent a sample "Level 2" reliability requirement. In other words, if the failure rate of the system is within the Level 2 bounds, the system can be labelled as "Level 2 Reliable."

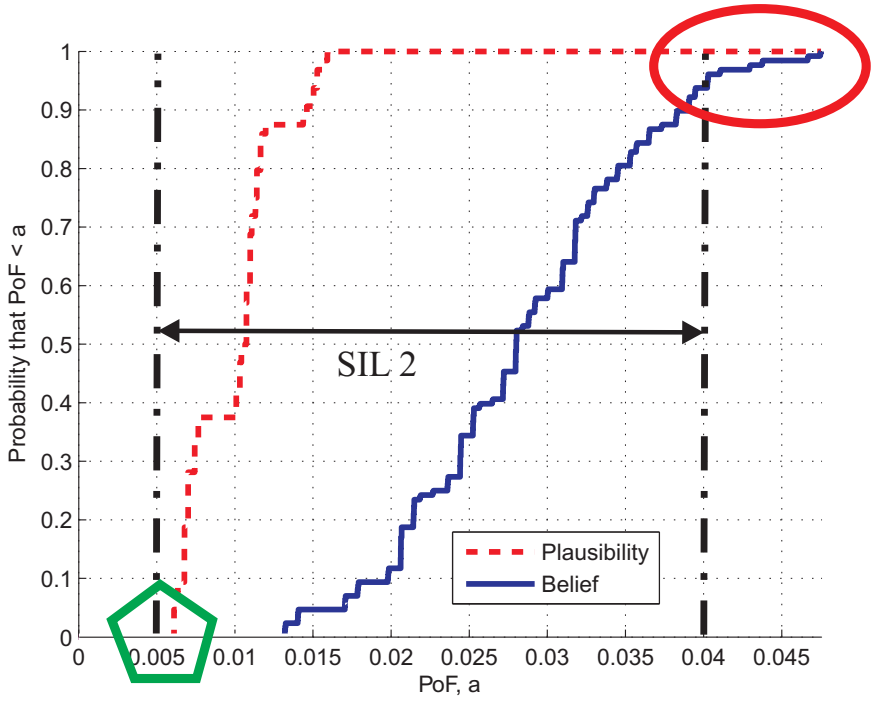


Figure 20: System BPA for NSL's Rascal UAV

The green pentagon shows where *Plausibility* and *Belief* functions intersect on the horizontal axis. It should be interpreted that there is absolutely no evidence (by any combination of upper or lower bound estimates) that the system's PoF is less than 0.005. Because the Level 2 lower bound is at 0.005, this means we have no evidence (and 0% probability) that the PoF is less than this value, so we cannot conclude the system's reliability is within the

Level 2 category $[0, 0.005]$.

The Level 2 upper bound is of greatest interest in this example. We can see from the red ellipse that the *Belief* and *Plausibility* functions do not meet until after the Level 2 upper bound, so there is some evidence that says the true failure rate could be higher than the Level 2 upper bound. If both *Plausibility* and *Belief* functions had reached 100% on the left side of the Level 2 upper bound, we could conclude with 100% confidence that the true failure probability is less than the Level 2 upper bound, and our system's reliability does in fact fit within the Level 2 category.

The greatest benefit from these plots is a qualitative assessment of reliability. At the Level 2 upper bound, the *Plausibility* function has already reached 100% meaning there is plausible evidence that says the failure rate meets SIL 2 criteria. The *Belief* function, while not at 100%, is roughly 94%. It should be interpreted that 94% of the evidence suggests the failure probability meets SIL 2 criteria. Because we have such large percentages of evidence at both Belief and Plausibility functions at the Level 2 upper bound, it could be concluded that the NSL Rascal UAV does in fact belong within the Level 2 Reliability category.

3.8 Evaluating Reliability & Uncertainty From a BPA Plot

For the system reliability to be evaluated iteratively through improvements, we must evaluate it at a single point for comparison before and after improvements. Given some degree of confidence (90%, 95%, etc.), we must choose a single point between the *Plausibility* and *Belief* curves at which we evaluate the system's PoF. We must first choose a confidence level (90%, 95%, etc.). This level dictates where, vertically, on the BPA we choose the *Belief* and *Plausibility* points. 100% confidence (while undefined in conventional probabilistic modelling) corresponds to the very top of the *Plausibility* and *Belief* curves – the best and worse case PoF from *all* of the evidence. For any confidence level between 0% and 100%, the BPA plot shows two distinct values: *Plausibility* and *Belief*. We must choose where, horizontally, we want to evaluate the PoF. We define $Risk$ as the percentage of risk we're willing to take

– 0% *Risk* is the *Belief* value (worse case PoF), and 100% *Risk* is the *Plausibility* value (best case PoF). 50% *Risk* is midway between *Belief* and *Plausibility*. Simply put, this variable defines how optimistic we are about the data presented – do we want to choose more of a best case scenario or worst case?

Intuitively, epistemic uncertainty can be judged by the width of a focal element – the wider the range of PoF, the more uncertainty is associated with it. By extension, the "average width" between the *Belief* and *Plausibility* curves on a BPA plot represents the amount of uncertainty in the PoF – the greater the distance between *Belief* and *Plausibility*, the greater the uncertainty.

To evaluate a source or system for its amount of epistemic uncertainty, the BPA "Average Width" (AW) can be calculated by summing the $range * mass$ of each focal element in the final system DSV. Systems with large uncertainties have *Plausibility* and *Belief* curves relatively far apart, while systems with zero uncertainty (deterministic systems) have coincident *Plausibility* and *Belief* curves (on top of each other). In the sample Rascal UAV model (Figure 20), 10 components each having 2 failure data sources generates a system DSV with 1,024 focal elements. The AW is simply the product of each range and its mass, for each DSV. Figure 21 illustrates the "width" between a point on the *Plausibility* and *Belief* functions. The AW is simply the average of all widths.

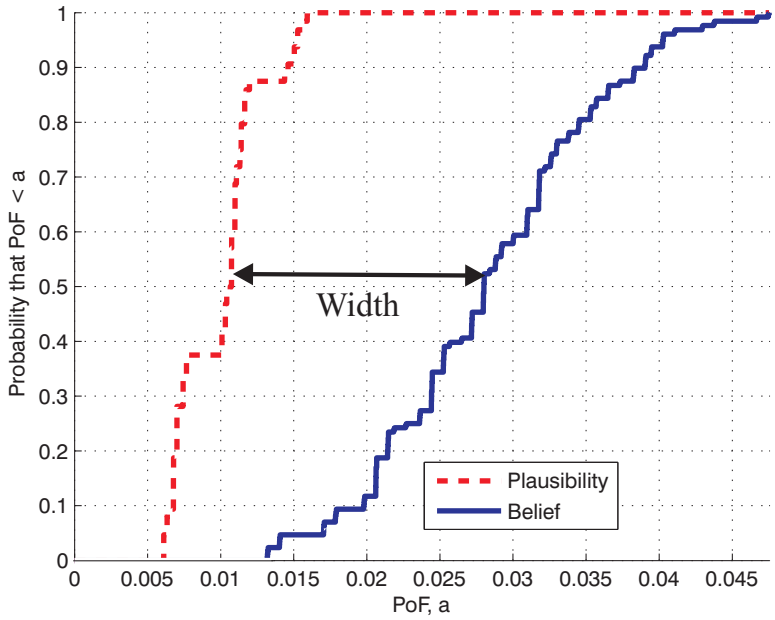


Figure 21: Width of Two Points on the NSL Rascal's BPA Plot

4 Sensitivity Analyses

Sensitivity analysis, in general, is a systematic study of how varying inputs affects various outcomes of the model. Here, we're concerned with reducing the system's PoF and PoF uncertainty (as measured by the AW of the BPA plot) by researching or improving parts within the system. These analyses are critical to cost-effectively improve the reliability of a system, as discussed in Section 2. Once we've found a baseline PoF and AW for our system, these analyses highlight the component that's most cost-effective to improve.

To improve a system cost-effectively, one must identify the components that have the largest effect on the system's PoF or uncertainty (whichever is deemed to be the most critical issue). Further, improvement cost must be incorporated into this analysis because often the best thing to improve is not the cheapest thing to improve.

Once a baseline system BPA is developed (as in Figure 20) we may want to improve the (estimated) reliability by finding more accurate data on current components, swapping to

higher quality parts, or adding redundant parts. We could 1) research a current component to reduce the uncertainty in its PoF estimates, 2) swap that component for a higher quality alternative, or 3) add a twin redundant part. Because we could improve the estimated reliability with any of these methods, finding the most cost-effective route requires solving three sensitivity analyses: 1) sensitivity to data uncertainty, 2) sensitivity to higher quality parts, and 3) sensitivity to component redundancy.

4.1 Sensitivity Analysis #1: Sensitivity to Improved PoF Data

Previous research has been done in regards to sensitivity analysis with Dempster Shafer structures by Ferson et. al [10]. Sensitivity analysis involving epistemic uncertainty can be performed by "pinching" a focal element (or entire DSV). With this new "pinched" DSV for a given component's PoF, the system can be re-analyzed to obtain a new AW. Reducing the PoF uncertainty for any given component will reduce the PoF uncertainty for the system. To ensure a *cost-effective* analysis, the percent difference between the "unpinched" and "pinched" AW values should be divided by the improvement allowance (referred to as Money Allocated To Improve, $MATI$). For example, if pinching a servo's and an autopilot's DSVs both results in an equivalent reduction in system uncertainty (AW), the estimate of the servo's reliability should be improved first, as it presumably costs less to perform accelerated life cycle testing on a servo than to assess reliability of an autopilot.

The pinch magnitude (noted as $PercentUncertReduction$) is up to the user. It is most intuitive to choose some value that is feasible and likely. If I spend X dollars to research a component or perform accelerated life tests on it, what percent reduction in PoF uncertainty can I reasonably expect? This idea of "assumed uncertainty reduction" is somewhat abstract, but it is reasonable to assume that if we spend money researching a component we'll learn something about it, allowing us to narrow down the PoF range. Figure 22 illustrates this "pinch" method on a sample DSV with a $PercentUncertReduction$ of 50%.

To complete the sensitivity analysis, the following steps should be performed:

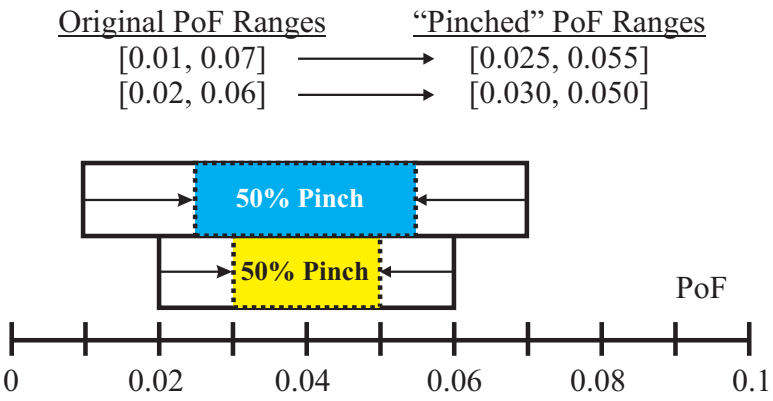


Figure 22: Sample DSV "Pinch" for a Given Component with Two Sources of PoF Estimates

1. Evaluate original system AW (defined in Section 3.8).
2. Temporarily pinch a component's DSV range by $PercentUncertReduction$ about the DSV midpoint.
3. Evaluate system AW again.
4. Compute percent difference between original system AW and revised AW with the improved component PoF data.
5. Divide this percent difference by the improvement allowance ($MATI$).
6. Reset this component's PoF DSV to the original value.
7. Repeat steps 1-6 for *each* component in the Fault Tree.

After progressively pinching each component's uncertainty (AW), the sensitivity analysis yields a bar graph showing percent reduction in system AW per dollar spent for each component in the system. The process identifies the component that reduces the system AW the most *per dollar spent to research it.* The PoF estimate for this component should be improved first to reduce the epistemic uncertainty of the system in the most cost-effective manner. Figure 23 shows the uncertainty sensitivity analysis results for the Rascal's Pitch Control subsystem; the Fault Tree for this subsystem was given in Figure 13

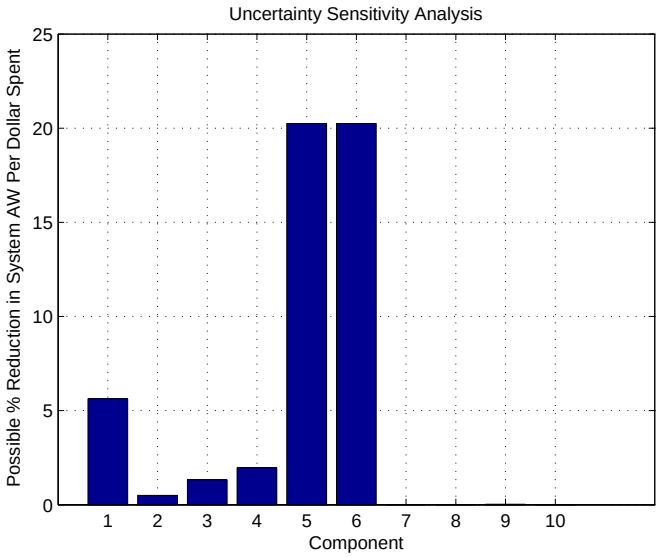


Figure 23: Uncertainty Sensitivity Analysis for the Rascal UAV's Pitch Control Subsystem

Reducing the PoF uncertainty for a component is relatively straight-forward for a component with a single data source. The process can proceed similarly if there are multiple sources for a given component's PoF. For example, in step 2 of the procedure above, one could pinch all focal elements for the component (as in Figure 22, effectively pinching all sources of evidence at once. With a given component's AW pinched, the system AW is evaluated again to find the % difference (per dollar) due to the given component's "reduced uncertainty."

The decision to pinch all focal elements in equal percentages is somewhat arbitrary. The "pinch location" has a slight effect on system AW. It is impossible to predict though where more research will shift the new PoF range, giving credence to a "random" pinch location. Again, random pinch locations will lead to varying outcomes which will make a direct sensitivity analysis difficult – pinching only the left or right sections of an estimate changes its center of mass, passively changing its reliability. However, if a consistent pinch location is chosen within each range, the ratio between improvements remains constant. Pinching each source to its midpoint is a consistent method to complete the sensitivity analysis.

4.2 Sensitivity Analysis #2: Sensitivity to Improved Components

To improve the reliability of a system with a given Fault Tree structure (e.g., without adding redundant parts) we could simply swap out a component for a higher quality (but more expensive) alternative. For example, if the original system had a basic servo, perhaps we can assume upgrading to a ball-bearing servo or metal gear servo would improve it's reliability and reduce the PoF. To pick the most cost-effective component to improve, a sensitivity analysis should be performed on all components.

For a component with a single focal element (a single PoF range), the DSV can simply be shifted by a small amount. We define this shift magnitude as $PercentPoFReduction$. This shift percentage, similar to the "pinch" percentage from Section 4.1, is the percent we assume the component's PoF will be reduced if we invest in a higher quality part. Figure 24 shows a 20% PoF improvement for a sample DSV.

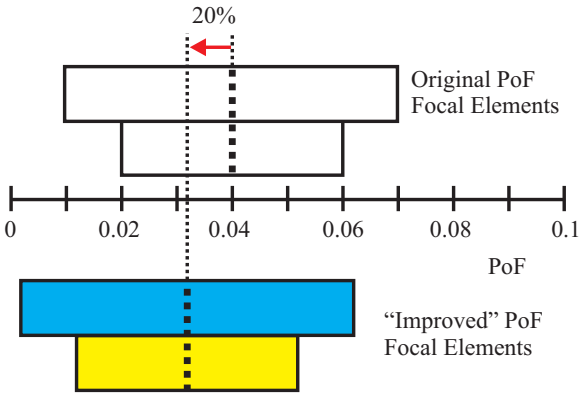


Figure 24: Sample DSV Shift

To calculate *where* to evaluate the actual shift, we can take this about any point within the range, as long as the location is consistent between components. For consistency, the "mid-point" (the weighted average PoF value, or "center of mass" of the DSV entries) between upper and lower bounds ($Belief$ & $Plausibility$) is chosen.

To complete the sensitivity analysis, the following steps should be performed:

1. Evaluate original system PoF about the $EvalPoint$ (defined in Section 3.8).
2. Shift a component's DSV by $PercentPOFReduction$ about the midpoint.
3. Evaluate the system PoF again after the shift.
4. Compute percent difference between original system PoF and this new "improved" system's PoF.
5. Divide this percent difference by the money allocated to improve/swap the component ($MATI$).
6. Reset this component's failure data with the original data.
7. Repeat steps 1-6 for *each* component in the Fault Tree.

This process results in a bar graph yielding the "percent improvement in system reliability per dollar spent" for each component. Simply stated, the system reliability can be improved most effectively if we replace the component selected by this sensitivity analysis with a better, higher quality part.

4.3 Sensitivity Analysis #3: Sensitivity to Redundant Components

While Section 4.2 focused on "swapping" the component for a higher quality alternative, this section introduces another way to increase reliability: add a redundant part for any given component. We saw in Section 2.2 (Equation (1)) that adding a redundant part increases reliability because *both* components must now fail for the subsystem to fail. Adding additional parts increases system cost and weight, so this is not always feasible; therefore, cost must be factored into the sensitivity analysis.

This sensitivity analysis adds a theoretical twin (redundant) part to each component, in turn, and calculates the reduction in system PoF. Figure 25 illustrates this; a single component

(A) from the original system (Figure 25a) is modified to act in a subsystem with a twin sibling. Figure 25b shows the new system with added redundancy.

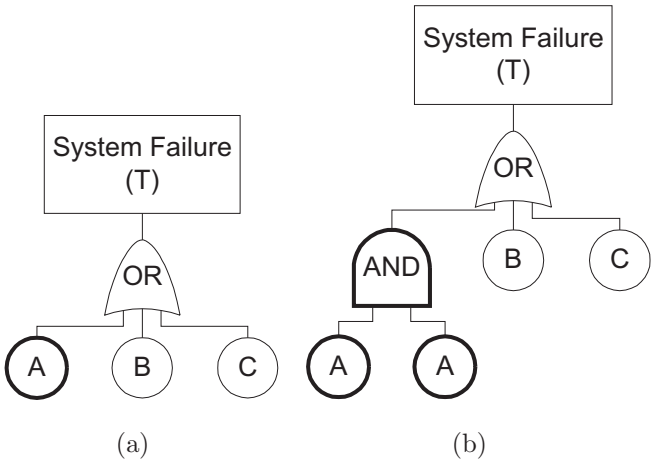


Figure 25: Adding Redundancy to Component A for Sensitivity Analysis #3

Different components have varying costs so the analysis must account for this; for instance, adding an additional autopilot costs a great deal more than an additional control surface, antenna, servo, etc.. To account for this, the *upfront* cost of the component is doubled to account for this new sibling. It can be argued that adding redundancy has more adverse implications on the system than just monetary cost – weight and structural requirements are increased. Therefore, *upfront* cost should include monetary value *and* cost from weight increases due to installing this component.

The sensitivity analysis is outlined below:

1. Evaluate original system PoF about the $EvalPoint$ (defined in Section 3.8).
2. Substitute this new subsystem into the system model in lieu of the original (single) component.
3. Compute the percent difference between the original system's PoF and the "improved" PoF.

4. Divide this percent difference by the original component's upfront cost; we have to buy an additional part because two are required instead of one.

5. Reset this component's failure data with the original data (single part).

6. Repeat steps 1-6 for *each* component in the Fault Tree.

Figure 26 shows sample sensitivity analysis results from Sections 4.2 and 4.3 on a single bar graph. We see from Figure 26 that it is most cost-effective to add redundancy to Component

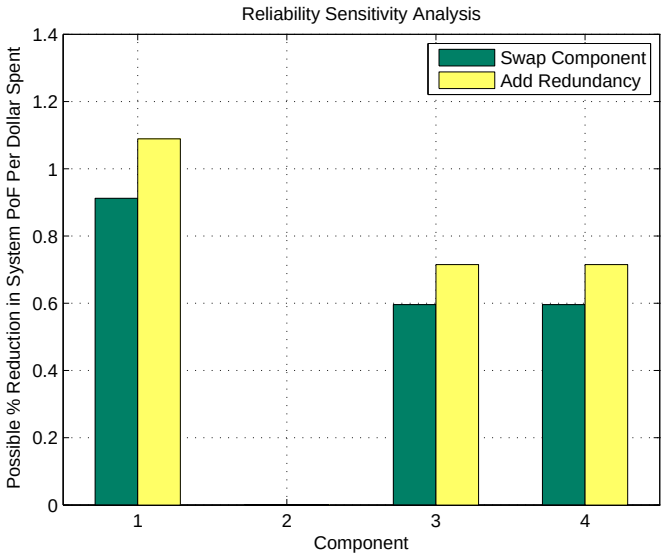


Figure 26: Sensitivity Analysis Results: Improved Part & Added Redundancy

1 for this sample system. If adding redundancy to Component 1 isn't feasible, the next most cost-effective improvement it to exchange Component 1 for a higher quality alternative part.

5 Structural Reliability

While PoF may be somewhat straight-forward for mechanical systems (servos, linkages, etc.), a platform's structure also plays into the (un)reliability of SUAV systems. It's simply not as easy to survey "experts" to find PoF ranges for structures. While other COTS parts are commonly used in a relatively predictable environment and sized for fairly predictable loads (i.e. servos, linkages, controllers, etc.), a structure's reliability can vary widely depending on many more factors. Simply put, we cannot just "estimate" a PoF of a structure without knowing intimate details of it's strength capacity or forces acting on it. Therefore, this section introduces ways of calculating PoF ranges from some predicted outside forces and the structure's strength – often well known data.

5.1 "Factors of Safety" and Limitations

Classical structural design consists of engineering a structure to withstand predicted forces. Often, either a "factor of safety" or "load factor" are introduced to bias strength requirements to allow for unknown or unpredictable forces. A common method used to size the required strength of a structure is shown in Equation (38).

$$\sigma_{yield} > \sigma_{max} * \text{FoS} \tag{38}$$

Here, the failure stress of the structure is σ_{yield} and the largest expected stress acting on the structure is σ_{max}. Strength and load variables are assumed to be deterministic, and factor of safety measures are defined by experimental observation, previous experience, economic concerns, or political considerations [20]. It is then assumed that the structure will not fail, given the predicted loads and factor of safety are sized correctly to withstand any likely force.

Intuitively, lowering the factor of safety will increase the probability of failure; unknown forces could exceed predicted forces due to a reduced factor of safety. To reduce weight and

cost, factor of safety should be kept small; to reduce the probability of failure, factor of safety should be inflated. The natural question becomes – how small a factor of safety is too small?

An alternative approach consists of evaluating a system's probability of failure based on imprecise (non-deterministic) strength and loading values. Using continuous functions for distributions allows a bit of uncertainty to be introduced. We can assume a structure's strength capacity and loading force distributions, predict the structure's PoF, and increase or decrease the structure's strength based on our evaluation of PoF. To reduce the PoF, we can modify or add to the structure; reducing structure (and therefore system weight and cost) will increase the PoF. This method aids in reliability-based structural design without requiring some assumed "factor of safety."

5.2 Gaussian Force Distributions to Calculate Probability of Failure

In this section, we investigate the PoF of a structure given Gaussian distributions for it's strength capacity (or structural resistance), R, and an outside force (load) acting on the structure, S. Gaussian (normal) distributions are commonly used when the shape of the true distribution is unknown. Figure 27 shows two sample Gaussian distributions.Here, R is the resistance (structural capacity) of the structure; S is the outside force acting on the structure. Intuitively, failure occurs when the outside forces are greater than the structure's strength – when $S > R$, or when $R - S < 0$. We can introduce a new random variable Z defined as the difference between resistance and outside force: $Z = R - S$. Therefore, the structure fails when $Z < 0$, or with probability: $P(Z < 0)$. Addition (subtraction) of any two random variable distributions can be computed with a convolution integral, as explained in [11]. With the assumption of Gaussian distributions, independence, and the relationship

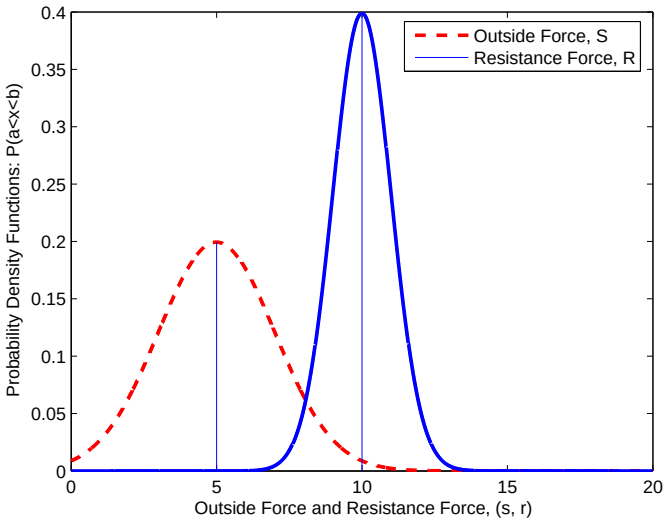


Figure 27: Sample Gaussian Distributions

$Z = R - S$, the convolution integral simplifies to:

$$\mu_Z = \mu_R - \mu_S \tag{39}$$

$$\sigma_Z^2 = \sigma_R^2 + \sigma_S^2 \tag{40}$$

where μ is the distribution's mean value, σ is the standard deviation. Recall that Z is the new random variable distribution that represents the difference in structural strength capacity and outside forces, and the probability of failure is simply $P(Z < 0)$. Therefore, PoF is represented by the area to the left of the origin in the resulting probability density function of Z, shown in Fig. 28.

To calculate this failure region's area (and thus the structure's PoF) we simply evaluate the cumulative density function (CDF) of Z at the origin, yielding $PoF = P(Z < 0)$. While a simple closed form function for a non-standard CDF does not exist, it can be solved numerically, with MATLAB's `normcdf`$(0, \mu_Z, \sigma_Z)$ function, or by using a standard lookup table. Melchers [20] suggests that a CDF lookup table from any statistics book can also be used:

$$PoF = \Phi\left(\frac{0 - \mu_Z}{\sigma_Z}\right) = \Phi\left[\frac{-(\mu_R - \mu_S)}{(\sigma_S^2 + \sigma_R^2)^{1/2}}\right] \tag{41}$$

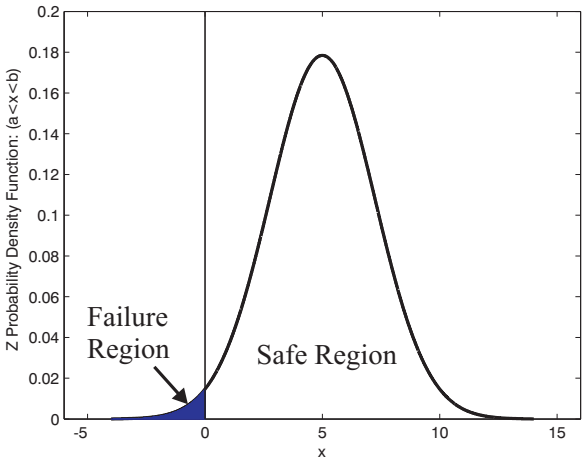


Figure 28: Structural Failure Region on Z

where $\Phi(-\beta)$ is the standard normal distribution function (zero mean and unit standard deviation).

For the above sample parameters, the structure's $PoF = 0.01267$. Using the conventional deterministic approach with $\mu_S = 5$ and $\mu_R = 10$, this yields a factor of safety of 2. While deterministic force calculations and a factor of safety suggest the structure should never fail under the predicted (deterministic) loads, we can calculate a structure's (non-zero) PoF if we introduce some uncertainty by assuming Gaussian force distributions for external loads and the structure's strength.

5.3 Imprecise Force Estimates

While the above section discusses ways to calculate PoF for stochastic systems based on deterministic mean values and standard deviations, these force distribution parameters may not be known with acceptable confidence. Many times, engineers use crude approximations for forces in the early design stages. Cost constraints can also limit an engineer's ability to model a system with high fidelity, so one must settle for a range of values for all forces present. Suppose we have maximum and minimum values for both resistance and outside forces acting

on the system. The inputs then have maximum and minimum values: $S = (\mu_S, \sigma_S)$ where $\mu_S \in [\mu_{S_{\min}}, \mu_{S_{\max}}]$ and $\sigma_S \in [\sigma_{S_{\min}}, \sigma_{S_{\max}}]$. Likewise, $R = (\mu_R, \sigma_R)$ where $\mu_R \in [\mu_{R_{\min}}, \mu_{R_{\max}}]$ and $\sigma_R \in [\sigma_{R_{\min}}, \sigma_{R_{\max}}]$.

The *lowest* probability of failure occurs when the structure's strength is much larger than outside forces, and the *highest* probability of failure occurs when they are closest together. All other combinations must be bounded within this PoF range. Put mathematically, these combinations become:

$$R = (\mu_{R_{\max}}, \sigma_{R_{\max}}), S = (\mu_{S_{\min}}, \sigma_{S_{\min}}) \rightarrow PoF_{\min} \tag{42}$$

$$R = (\mu_{R_{\min}}, \sigma_{R_{\min}}), S = (\mu_{S_{\max}}, \sigma_{S_{\max}}) \rightarrow PoF_{\max} \tag{43}$$

and the procedure from section 5.2 can be applied to solve both cases. This results in the range $[PoF_{\min}, PoF_{\max}]$.

For illustration, let's assume we estimate μ and σ for S and R to be:

$$\mu_S \in [4, 6]$$

$$\sigma_S \in [2, 2]$$

$$\mu_R \in [8, 12]$$

$$\sigma_R \in [1, 1]$$

(Assume all parameters have been non-dimensionalized.) Using the combinations from Equations (42)-(43), we find:

$$R = (\mu_{R_{\max}}, \sigma_{R_{\max}}) = (12, 1), S = (\mu_{S_{\min}}, \sigma_{S_{\min}}) = (4, 2) \rightarrow PoF_{\min} \tag{44}$$

$$R = (\mu_{R_{\min}}, \sigma_{R_{\min}}) = (8, 1), S = (\mu_{S_{\max}}, \sigma_{S_{\max}}) = (6, 2) \rightarrow PoF_{\max} \tag{45}$$

Using the approach from Section 5.2 (Equations (39)-(40)), we can plot the acting force, S, resistive force, R, and difference Z for both cases, PoF_{min} and PoF_{max} (Figure 29).

The resulting PoF range is $[0.000173, 0.185547]$.

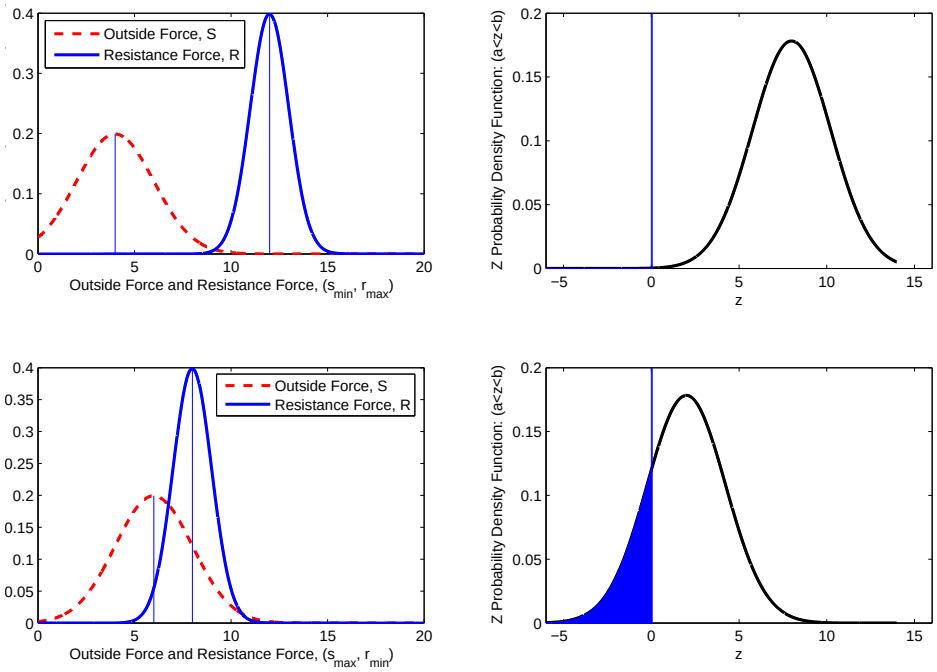


Figure 29: Distributions for S, R, and Z for Best (top) and Worst (bottom) Cases

In this section we have developed a way to calculate PoF directly from a structure's strength and the forces acting on it, rather than simply overbuilding the structure and *assuming* it's strong enough to never fail. Section 5.2 allows us to calculate PoF from any physical quantity (force, moment, stiffness, etc.), and Section 5.3 proves we can calculate a bounded range for PoF given imprecise inputs. Appendix A shows an example of this method as a cost saving measure to estimate the reliability of a strut-braced wing.

6 A MATLAB® Based "Evidence Theory / Fault Tree Analysis Toolbox" (ETFTA Toolbox)

While the results (PoF and cost increase) of the step-by-step reliability improvements (Section 2.3) are fairly easy to keep track of by hand, it can be exhaustive to calculate the system PoF and run sensitivity analyses for increasingly complex fault trees. A MATLAB® toolbox called the "Evidence Theory / Fault Tree Analysis Toolbox" (ETFTA Toolbox) was created to aide in these steps. One can simply input information about the system's components (cost, description, PoF, and $MATI$) and build the Fault Tree by defining which components are related in minimum-cut-sets (discussed in Section 6.1.3). The toolbox then generates the system's PoF, cost, AW, and BPA plot. In addition, the toolbox runs the three sensitivity analyses defined in Section 4 to highlight the most cost-effective improvements.

6.1 Data Input Formats

6.1.1 Analysis Parameters

The first step to writing an ETFTA m-file is to define the variables `Confidence` and `Risk` as global variables. This allows all sub-functions to use these values without repeatedly sending them through functions as arguments.

```
global Confidence Risk
```

Next, we must assign values to these variables. Recall, `Confidence` is the percentage of evidence we want to use when evaluating the system (Section 3.8). It defines where, vertically, on the BPA plot we choose the PoF location. `Risk` defines where, horizontally, on the BPA we choose the PoF location. It defines the horizontal placement between the *Belief* and *Plausibility* where we choose the PoF.

```
% Confidence level:  What percent of evidence do you want to represent the
% model?  90%?  95%?  100%?   (enter 0 to 1)
Confidence=0.95;

% Risk:  How much do you want to trust the Plausibility function?  0
% ignores plausibility, represents the model ONLY from Belief function (most
% conservative).  1 represents the model from the Plausibility function (most
% aggressive, risky assumption)  (enter 0 to 1)
Risk=0.10;
```

Next, we must define parameters for the sensitivity analyses. `PercentUncertReduction` is our assumed percent reduction in component AW if we spend money to research the component. `PercentPoFReduction` is the assumed percent reduction in component PoF if we swap for a better component (shifted at the midpoint of the DSV range).

```
% PercentUncertReduction:  Assuming more research is done on a component,
% how much can you assume the uncertainty will be improved by?
% (enter 0 to 100)
PercentUncertReduction=20;

% PercentPOFReduction:  Assuming we swap a component for a higher quality
% alternative, how much can you assume the PoF will be improved by?
% (enter 0 to 100)
PercentPOFReduction=30;
```

6.1.2 Component Data

Next, component data must be defined. In the m-file, each component is treated as a structure with fields `dsv`, `cost`, `description`, and `MATI`. To define a DSV, we use the format `[PoFlow, PoFhigh, mass]`. Note that the ETFTA Toolbox is designed to accept PoF *per*

hour (or simply "expected failures per hour"). Below is a sample DSV definition (line 1) and field assignment (line 2).

```
Source1=[0.01, 0.02, 1];
ComponentA.dsv=Source1;
```

If we have two or more focal elements (i.e., two or more PoF sources), we can combine them into a single DSV using the `combine([Source1,Source2])` function[6]. The parameter `mass` can be set to any relevant value; this weight is normalized among all relevant DSV entries in the function `combine`. Below is a sample input where we have two expert opinions:

```
Source1=[0.01, 0.02, 2];
Source2=[0.005, 0.03, 1];
ComponentA.dsv=combine([Source1,Source2]);
```

Note that we trust Source 1 twice as much as Source 2 so we set it's mass to be twice as large. The `combine` function automatically normalized these weights.

Finally, `cost`, `description`, and `MATI` must be defined as shown in the sample below. Values can be whatever is relevant, but `ComponentA`'s structure must have these fields.

```
ComponentA.cost=100;
ComponentA.description='Component A Description Here';
ComponentA.MATI=20;
```

For structural components, we define the external and resistive forces, S and R, respectively, rather than DSV ranges. The function `structurePoFcalc` is used by defining ranges for both S and R ($\mu_{S\text{high}}$, $\mu_{S\text{low}}$ for example) instead of DSV ranges. The code below shows a

[6]Note that only two sources were combined for this sample case, but `combine` can combine any number of sources.

sample input for a wire.

```
% S is the external force, R is the resistance (strength) of the structure
%XXX=structurePoFcalc([muSLow, sigmaSLow; muSHigh, sigmaSHigh],...
% [muRLow, sigmaRLow, muRHigh, sigmaRHigh]);
TensileForces=structurePoFcalc([100, 20; 120, 20],[300, 75; 400, 75]);
Wire.dsv=TensileForces;
Wire.description='Braided Wire';
Wire.cost=10;
Wire.MATI=5;
```

The ETFTA Toolbox includes an automated "Part Chooser" that runs a decision matrix and substitutes the "winner" into the component fault tree. This function is useful when there are many suitable components to choose from and we may want to consider more than just cost and reliability. (For example, perhaps we'd like to consider weight and size as well.) We must define a `weights` vector where we input our subjective weighting for each trait. The function is `ChoosePart` and is called in the following format:

`winningPart=ChoosePart([part1,part2],weightVector)`

The following shows sample inputs for two servos and the `ChoosePart` function:

```
Servo1Source1=[0.001, 0.00125, 1];
Servo1Source2=[0.00111, 0.001429, 1];
servo1.dsv=combine([Servo1Source1,Servo1Source2]);
servo1.description='Futaba S3152';
servo1.cost=35;         % dollars
servo1.MATI=10;         % dollars
servo1.weight=1.5;      % oz
servo1.torque=1/87;     % 1/(oz-in) at 6V
servo1.speed=1/353;     % sec/degree
servo1.volume=1.79;     % in^3
```

```
Servo2Source1=bpa([0.002, 0.004, 1]);
Servo2Source2=bpa([0.001429, 0.002,1]);
servo2.dsv=combine([Servo2Source1, Servo2Source2]);
servo2.description='Hitech HS-645';
servo2.cost=20;         % dollars
servo2.MATI=10          % dollars
servo2.weight=2.11;     % oz
servo2.torque=1/133.3; % 1/(oz-in) at 6V
servo2.speed=1/428;     % sec/degree
servo2.volume=1.82;     % in^3

% PoF, AW, Cost, MATI are always evaluated.  Other characteristics must be
% given weights as well. Weight vector must be defined as:
% [PoF_Weight, AW_Weight, Cost_Weight, MATI_Weight, other_weight1...]
% ServoWeights=[PoF, AW, Cost, Mati, weight, torque, speed, volume]
ServoWeights=[10,3,10,4,2,4,1,1];
ServoA=ChoosePart([servo1,servo2],ServoWeights);
```

We can define any number of traits with each component option, as long as each option has the same trait names and they're in the same order. Notice in this sample input that we've weighted PoF and Cost relatively higher than any other traits. In the above sample, `servo1` wins the decision matrix and therefore ServoA takes on the properties of `servo1` in the fault tree (not shown).

6.1.3 Fault Tree Structure

The ETFTA toolbox doesn't accept visual fault trees, but rather a list of minimum-cut-sets (MCS) – a critical group of components or subsystems such that if any one of these MCS fails, the system fails.

To find these MCS from a fault tree, we progressively pick a component and trace its path to the top event. If it does *not* pass through any AND gates (as in component A in Figure 30b),

the component is itself a MCS. If, however, the component passes through an AND gate, then there is at least one additional component in the MCS. Components B and C are both inside MCS 2 in Figure 30c because *both* components B *and* C must fail for the top event to occur.

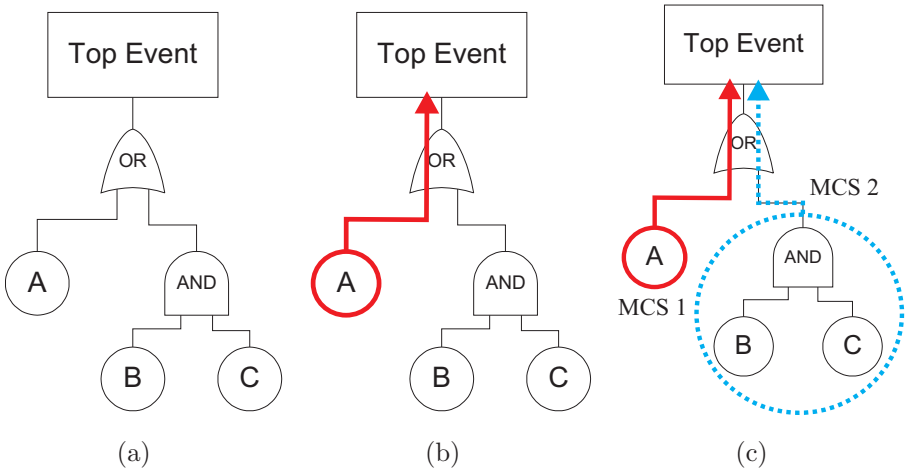


Figure 30: MCS Identification on a Simple Fault Tree

Thus, for the fault tree in Figure 30, there are two minimum-cut-sets; MCS 1 has component A and MCS 2 has components B and C. This *visual* approach to finding MCS is sufficient for fairly simple fault trees typically found in small UAV design.

Various methods have been proposed to calculate MCS for more complex trees using formal algorithms. Fard [9] and Kara-Zaitri [15] both propose efficient algorithms to calculate MCS. Barlow discusses the MOCUS (Method for Obtaining Cut Sets) method [2]. While their proposed algorithms could be useful, formalized fault tree evaluation programs are already available for commercial use. They are valuable when dealing with extremely complex systems where components appear in multiple places in fault trees. They allow a user to graphically draw a fault tree, assign labels to each component, and calculate which components belong in corresponding MCS. Popular programs include *Relex Fault Tree*® by Relex Software Corp, *FaultTree+*® by Isograph Inc., and *EventTree*® by Item Software.

Once the list of MCS has been completed, they can be input into the ETFTA Toolbox. For

the system in Figure 30, the MCS inputs are as follows:

```
MCS(1).component{1}=A;
MCS(2).component{1}=B;
MCS(2).component{2}=C;
```

This is equivalent as saying the first component in MCS 1 is component A. The first component in MCS 2 is component B. The second component in MCS 2 is component C. Note that lines 2 and 3 of this code define the AND gate in the fault tree, and line 1 defines MCS 1 as having a single component A. Appendix B includes an example of this MCS input for a more complex fault tree.

6.1.4 Run Commands

With the evaluation parameters and component data defined, we now focus on the run commands to execute the ETFTA analysis.

The first command that must be executed is the `conditionDSVs` function. It checks that the smaller PoF value is in fact on the left side of the DSV and the larger PoF value is in the middle of the DSV (with `mass` being the third term). If not, it switches the order of the PoF values. This function is useful because DSV entries can take intuitive ranges such as: `A.dsv=[1/500,1/600,1]` which says component A should fail sometime between 500 and 600 hours. As stated, however, the higher PoF value is listed first. The function `conditionDSVs` switches the order of these two entries automatically.

```
% Condition DSV entries so Plaus<Bel
MCS=conditionDSVs(MCS);
```

`systemDSV` aggregates all DSV entries into a single, system DSV. It uses the MCS structure to automatically calculate AND and OR gate interactions from the fault tree to generate

the system DSV.

```
% Run the system model
SysModel=systemDSV(MCS);
```

`systemPoF` calculates the *PoF* of the system and outputs estimated failures per 100,000 hours.

```
% Calculate "System PoF" of the Original system:
PoFSystem=systemPoF(SysModel);
fprintf('PoF for the system: %0.10f\n',PoFSystem)
fprintf('Failures:  %0.1f per 100,000 hours',PoFSystem*100000);
```

`systemCost` calculates the total upfront cost of the system.

```
% Calculate total cost of the system:
TotalSystemCost=systemCost(MCS);
fprintf('Total System Cost: %0.0f\n',TotalSystemCost)
```

`AW` calculates the *AW* of the system. Recall that *AW* is a metric for *PoF* uncertainty; it is the average distance between the *Plausibility* and *Belief* curves on the system BPA.

```
% Calculate "Average Width" of the Original system:
AWsystem=AW(SysModel);
fprintf('AW for the system: %f\n\n\n',AWsystem)
```

`plotBPA` plots the system BPA.

```
% Plot the new system BPA
```

```
plotBPA(SysModel, PoFSystem)
```

`uncertSense` runs the uncertainty sensitivity analysis defined in Section 4.1. This generates a bar graph showing the possible percent reduction in system AW (per dollar spent), if a given component's AW is reduced by an amount $PercentUncertReduction$.

```
% Run PoF uncertainty sensitivity analysis.
uncertSense(SysModel,MCS,PercentUncertReduction);
```

`PoFSense` runs the PoF reduction sensitivity analyses defined in Sections 4.2-4.3. This generates a bar graph showing possible percent reduction in system PoF (per dollar spent), if a) each respective component is swapped for a more reliable alternative or b) redundancy is added.

```
% Run PoF sensitivity analysis.
PoFSense(SysModel,MCS,PercentPOFReduction);
```

Finally, `PrintLegend` identifies each component from the sensitivity analysis graphs. The sensitivity analysis bar labels are numeric so this legend is used to correlate the numeric labels to each component's description. It also identifies which winning part was chosen if `ChoosePart` function was used.

```
PrintLegend(MCS)
```

6.2 Outputs

6.2.1 System Results and Legend

When the run commands from the previous section are executed, the following results and legend are displayed. Note the PoF, AW, Cost, and component data are contrived, but are used as place holders here to show formatting. Also, PoF data was given on a "per hour" basis (the probabilities were "probability of failure per hour of use") so the expected number of failures is given per 100,000 hours.

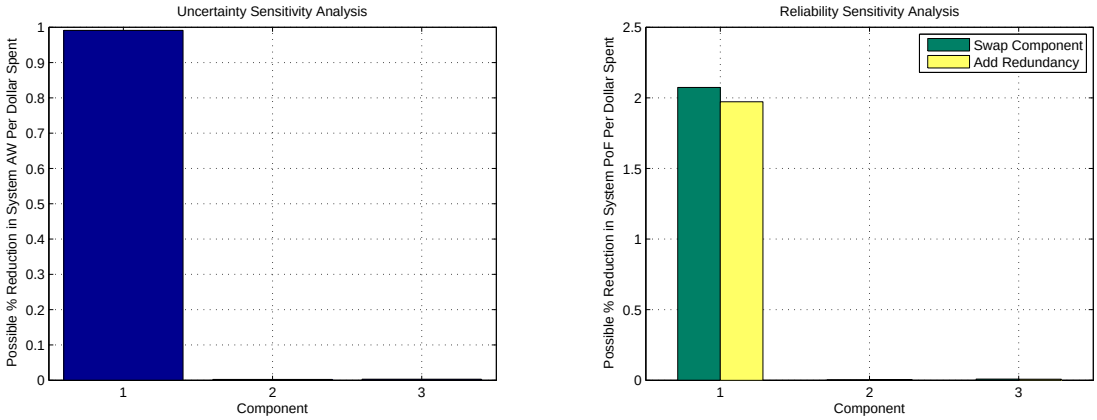
```
PoF for the system: 0.0090405950
Failures:  904.1 per 100,000 hours
Total System Cost: $210
AW for the system: 0.005045

---------------------- Sensitivity Legend --------------------------
  Part #                  Description          Cost($)      MATI($)
------------------------------------------------------------------------
    1                      Component A           50           20
    2                      Component B          100           50
    3                      Component C           60           20
------------------------------------------------------------------------
```

6.2.2 Sensitivity Analysis Results

Figure 31 shows the sensitivity analysis results for this sample case.

Notice components 2 and 3 on the sensitivity analyses have very little effect on the system – individually, components 2 and 3 have relatively little effect on the system PoF of PoF uncertainty because they are each part of a redundant sub-system (AND gate). We can use the legend from the MATLAB output (previous section) to see that components 2 and 3 correspond to components B and C which are related through an AND gate. Component 1's criticality makes intuitive sense because this is a redundant system where *both* components must fail for the system to fail. Therefore, either one of these components on its own has



(a) (b)

Figure 31: Sample Sensitivity Analysis Bar Graphs

little effect on the system's PoF or PoF uncertainty. Component 1 (A) has a much larger effect on the system PoF and PoF uncertainty because it is a single component MCS – failure of this single part will cause system failure.

Using the inputs and outputs discussed in this chapter, a user can easily create an m-file that contains all relevant failure data, the fault tree structure, and concise run commands to evaluate the system. The next chapter presents a case study where this toolbox is used extensively, and Appendix B presents a complete ETFTA m-file representing the SPAARO UAV's initial fault tree, failure data, and run commands.

7 CASE STUDY: "Designing in" Reliability into the SPAARO UAV

7.1 Motivation

Virginia Tech's Nonlinear Systems Laboratory (NSL) is committed to developing a fleet of fully autonomous UAVs to support graduate research and complement the undergraduate curriculum with a "real life" platform. Students in the NSL have previously modified off-the-shelf 110 inch wingspan Sig® Rascal airframes, but increased payload and fuel requirements forced the Rascals to be flown at 28lbs – roughly twice the weight they were designed for. Further, the airframes were suffering from structural fatigue because payloads were difficult to install in cramped quarters. Figure 32 shows a NSL Rascal UAV shortly before launch.

Figure 32: Rascal UAV on a Runway

The Rascal airframes were dangerously overweight, over-stuffed and obsolete. A graduate student and two undergraduates were tasked with developing a larger, more reliable airframe that would serve as the department's workhorse for years to come. The resulting UAV, "SPAARO" (Small Platform for Autonomous Aerial Research Operations), is described in more detail by Murtha et. al [22] and Cotting et. al [4] and is shown in Figure 33.

The following case study serves as a practical example for those who wish to apply the

Figure 33: SPAARO UAV on a Runway

methods described in this thesis. It illustrates the process of designing a more reliable airframe based on typical radio-controlled (r/c) components and configurations.

To keep cost and complexity down, r/c platforms typically don't have any built-in redundancy so the reliability of the system is dependent on any single component. The SPAARO UAV must be inherently reliable and must be capable of safely flying payloads for years to come. This case study modifies and improves a baseline r/c platform to be more reliable in a cost effective way.

7.2 Preliminary Design

Initial sizing was the first step to the design. As stated in [22], the UAV sizing was driven by performance requirements and a weight limit of 55 lbs. Sizing was frozen as shown in Table 6.

Students chose a pusher configuration with the engine mounted to the rear of the fuselage to allow a large payload bay in the UAV's nose. With sizing frozen, Figure 34 shows the

Max Weight	55 lbs
S	16 ft^2
Power	5.5 HP
$\mathcal{R}$	9

Table 6: Initial Sizing for SPAARO UAV

SPAARO's planform.

Once initial sizing was complete and a basic planform was chosen, students looked into reliability when designing control systems, surfaces, and structures. Figure 35 shows the fault tree for a basic r/c airplane that was chosen to start the design[7].

All computations for this case study are handled by the ETFTA Toolbox as described in Section 6. Recall that the ETFTA toolbox doesn't accept fault trees, but simply a list of minimum-cut-sets (MCS). For this simple example, we can visually inspect the fault tree to generate the list of MCS (shown in Table 7). Note that all components, except the aileron surfaces and servos, are single component MCS; there is no redundancy in this baseline design. (Ailerons do have some passive redundancy from symmetry.) Two UAV pilots with a combined 20 years of r/c experience were surveyed and failure data & costs were added to the component list (Table 7). Note component D (Pilot Error) is our attempt to include crashes caused by human error during piloting. The cost associated with this ($1600) is from estimated labor costs associated with pilot training ($20/hour for two work weeks). The $MATI$ for this component ($5000) was chosen to be the increase in cost to buy a simple autopilot – the next more reliable alternative to a human pilot.

The code below shows how component data from Table 7 was input into the ETFTA m-file for component A (the 72MHz receiver).

[7]Note that this fault tree is very basic. Dermentzoudis has developed more detailed fault trees for basic platforms [6].

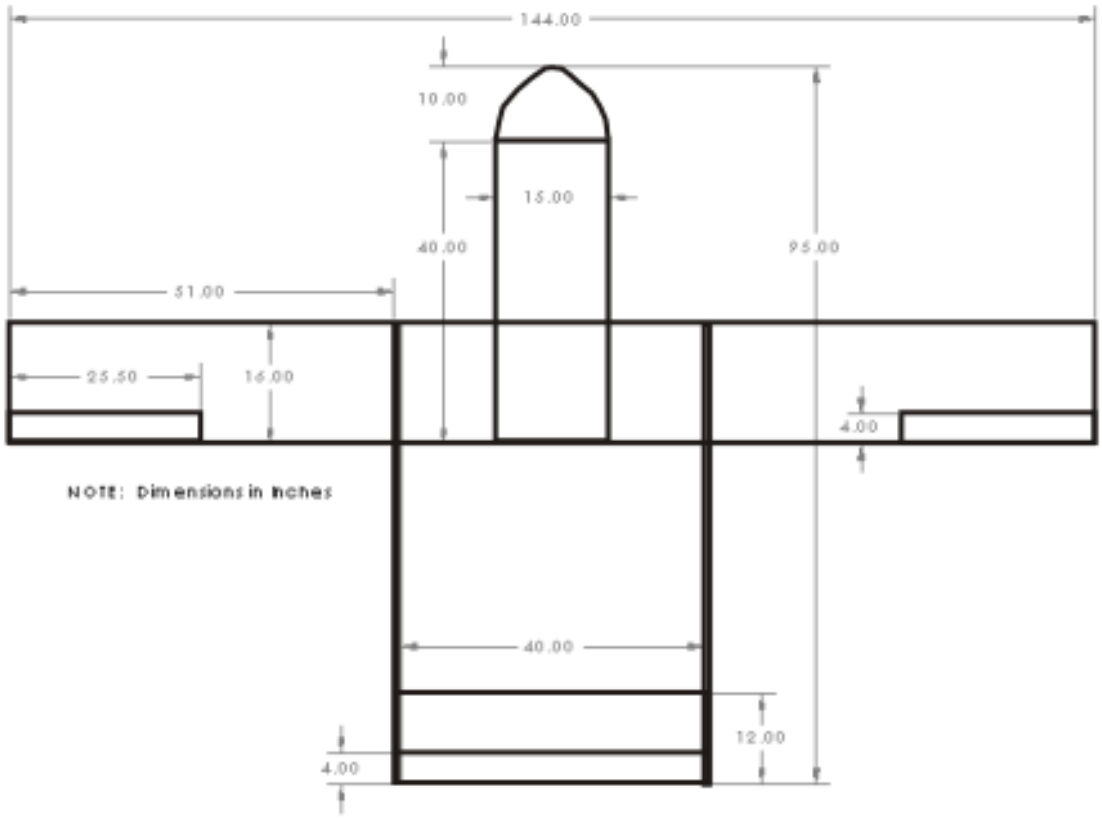


Figure 34: Sketch of Overall SPAARO Planform

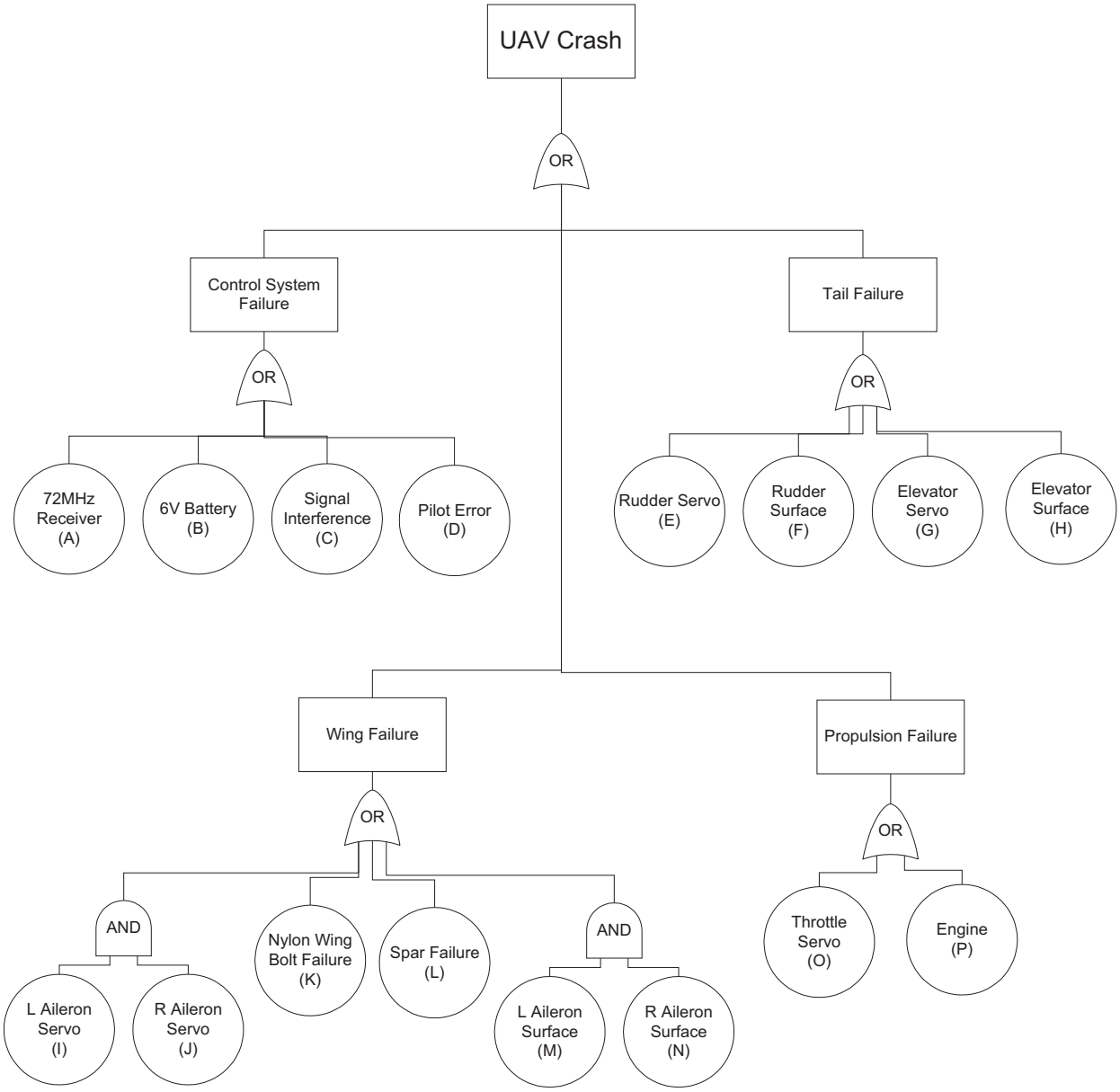


Figure 35: CASE STUDY: Initial Fault Tree

MCS	Label	Description	*PoF* Source 1	*PoF* Source 2	Cost ($)	*MATI* ($)
1	A	72MHz Receiver	{1e-5, 2e-5}	{1e-5, 2e-5}	$50	$100
2	B	6V Battery	{2e-4, 1e-3}	{1e-4, 2e-4}	$30	$30
3	C	Signal Interference	{5e-6, 1e-5}	{1e-5, 2e-5}	$1	$1
4	D	Pilot Error	{0.03, 0.05}	{0.01, 0.1}	$1600	$5000
5	E	Rudder Servo	{2e-3, 3.33e-3}	{2e-3, 3.33e-3}	$20	$15
6	F	Rudder Surface	{2e-4, 1e-3}	{1e-4, 2e-4}	$10	$10
7	G	Elevator Servo	{2e-3, 3.33e-3}	{2e-3, 3.33e-3}	$20	$15
8	H	Elevator Surface	{2e-4, 1e-3}	{1e-4, 2e-4}	$10	$10
9	I	Left Aileron Servo	{2e-3, 3.33e-3}	{2e-3, 3.33e-3}	$20	$15
	J	Right Aileron Servo	{2e-3, 3.33e-3}	{2e-3, 3.33e-3}	$20	$15
10	K	Nylon Wing Bolt			$5	$15
11	L	Main Wing Spar			$300	$200
12	M	Left Aileron Surface	{2e-4, 1e-3}	{1e-4, 2e-4}	$10	$10
	N	Right Aileron Surface	{2e-4, 1e-3}	{1e-4, 2e-4}	$10	$10
13	O	Throttle Servo	{2e-3, 3.33e-3}	{2e-3, 3.33e-3}	$32	$15
14	P	Engine	{0.05, 0.1}	{0.01, 0.05}	$500	$500

Table 7: MCS Assignments & *PoF* Data for SPAARO's Initial Fault Tree

```
% component A evidence:
JustinA=[1e-05, 2e-05,1];
% component A descriptions:
A.dsv=JustinA;
A.cost=50;
A.description='72MHz Receiver';
A.MATI=100;
```

Note the missing PoF values for components K and L in Table 7 (nylon wing bolt and main wing spar). The next step was to design the wing structure (specifically the main wing spar) and obtain PoF data. Because the outer 51 inch wing portions need to be removable, students chose a concentric circular tube design – the outer wing spar tubes would slide into a mating center wing tube. For these spars, students chose a 1.25 inch outer diameter (OD) and 1.125 inch inner diameter (ID) carbon fiber tube with a circular cross section.

First, we solve for b given $Æ$and S:

$$Æ = \frac{b^2}{S} \Rightarrow b = (+)\sqrt{Æ * S} \tag{46}$$

An elliptical lift distribution was assumed on the wing. Rather than dealing with this continuous force distribution, students calculated a point load[8] on the wing that would result in the equivalent bending moment at the root. Figure 36 illustrates this transformation; Figure 36a shows the continuous lift distribution and Figure 36b shows the equivalent point load. The spanwise location for this point load was calculated from Equation (47).

$$x_{\text{centroid}} = \frac{2b}{3\pi} \tag{47}$$

where x_{centroid} is the spanwise distance from root to the point load (lift centroid) on either wing and b is the full wingspan (both wings) calculated from Equation (46).

The bending moment at the root becomes $M = \frac{W}{2} * x_{\text{centroid}}$ where W is the total UAV weight. A consolidated bending moment equation given $Æ$ and S is therefore:

[8]The magnitude of this point load is equivalent to each wing's lift force.

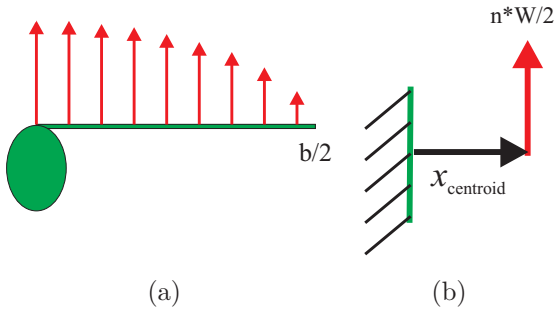


Figure 36: Continuous to Point Load Lift Transformation

$$M = \frac{nW\sqrt{\mathcal{R} * S}}{3\pi} \tag{48}$$

where n is the desired load factor and W is the total weight of the UAV.

Assuming a maximum load factor of 3g's for the SPAARO we have $M_{\text{nominal}} = 2521$ lbs/in. Undergraduate students ordered a sample spar tube and performed a bending moment test to find it's critical bending moment[9]. The experiment yielded a critical bending moment of 3600 lbs/in. Equation (48) can be manipulated to reveal the wing spar should nominally fail during a 4.25g maneuver.

Nylon wing bolts[10] have a minimum tensile strength of roughly 10,000 psi. For 1/4-20 bolts (as chosen for the main wing bolts) this corresponds to a maximum tensile load of 490 lbs per bolt. Because in practical use it's difficult to tell which of the four installed bolts are tightened the most or how much tensile force results simply from tightening the bolt, we use a conservative assumption that we have only *one* bolt taking loads for the main wing. Even with this assumption, the bolts should not fail before reaching an 8.9g load.

Having strength capacities for the main wing spar and nylon bolts, we can use the structural PoF calculation method discussed Section 5 and the ETFTA Toolbox to calculate PoF values from these strengths and estimated loads. Table 8 summarizes the relevant loads on

[9]That is, the moment required to delaminate and break the tube.

[10]Nylon 6/6 material.

the spar and bolt under a 3g maneuver[11].

Force	Mean Value, μ	Std Deviation, σ
S_{spar}(Bending Moment)	2521 lbs/in	200 lbs/in
R_{spar}(Bending Moment)	3600 lbs/in	500 lbs/in
S_{bolt}(Tension)	165 lbs	20 lbs
R_{bolt}(Tension)	490 lbs	100 lbs

Table 8: Summary of Forces for SPAARO's Initial Spar and Wing Bolt

To find the PoFs for these structural components, the command `structurePoFcalc` is used in the ETFTA input rather than PoF (DSV) inputs. The code below shows the sample input code for this command.

```
%sourceName=structurePoFcalc([muSLow, sigmaSLow; muSHigh, sigmaSHigh],...
% [muRLow, sigmaRLow, muRHigh, sigmaRHigh]);
JustinBoltForces=structurePoFcalc([165, 20; 165, 20],[490, 100; 490, 100]);
K.dsv=JustinBoltForces
K.description='Nylon 1/4-20 Wing Bolt';
K.cost=5;
K.MATI=15;
```

Note that we've calculated deterministic forces for the structure (Table 8) so the ETFTA toolbox force inputs are deterministic as well. (We've calculated the mean value of S to be 165 lbs with a standard deviation of 20 lbs, and the mean value for external loads is 490 lbs with a standard deviation of 100 lbs.) We could generalize S or R forces and define them as ranges to allow for a bit more uncertainty (perhaps if we don't have the human resources to do an in-depth analysis to calculate the strength of our structure). This issue is discussed extensively in Appendix A. Note also the standard deviations in Table 8. These deviations

[11]Recall S is the acting load and R is the resistive strength of the structure.

are important for predicting PoF (recall that $\sigma = 0$ results in 0% PoF if $\mu_R > \mu_S$).

7.3 Reliability Modelling and Improvement

Once preliminary design is complete and failure data is gathered for all components, we can run the ETFTA Toolbox to evaluate the design's (un)reliability and see which components are most cost-effective to improve. Appendix B shows the complete inputs and run commands from this initial ETFTA Toolbox run. The MATLAB output from this baseline design is shown below:

```
PoF for the system: 0.1953100551
Failures:  19531.0 per 100,000 hours
Total System Cost: $2626
AW for the system: 0.095797

---------------------- Sensitivity Legend --------------------------
  Part #             Description                Cost($)      MATI($)
------------------------------------------------------------------------
    1                    72MHz Receiver           50          100
    2                        6V Battery           30           30
    3               Signal Interference            1            1
    4                       Pilot Error         1600         5000
    5                     Rudder Servo            20           15
    6                    Rudder Surface           10           10
    7                    Elevator Servo           20           15
    8                  Elevator Surface           10           10
    9                 Left Aileron Servo          20           15
   10                Right Aileron Servo          20           15
   11             Nylon 1/4-20 Wing Bolt           5           15
   12           Main Wing Spar - Carbon          300          200
   13              Left Aileron Surface           10           10
   14             Right Aileron Surface           10           10
   15                    Throttle Servo           20           15
   16                            Engine          500          500
------------------------------------------------------------------------
```

Note that the predicted failure rate is extremely high – nearly 20%. The system's BPA is shown in Figure 37.

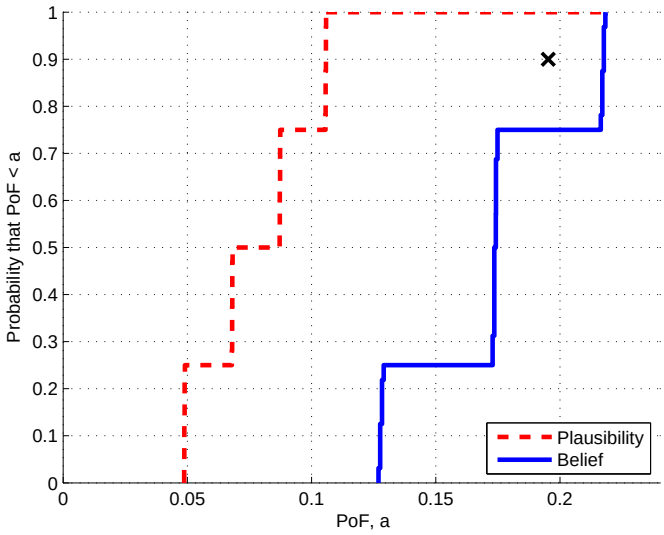
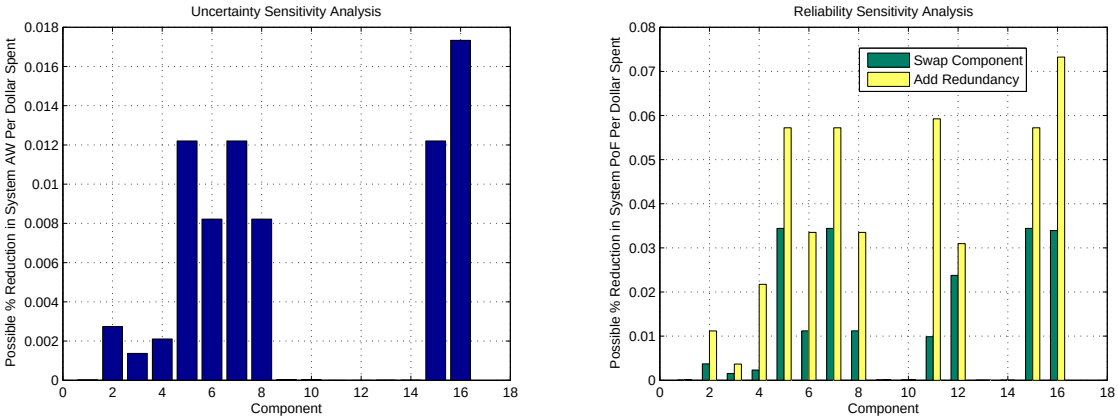


Figure 37: SPAARO BPA: Initial Design

The measure of the system's uncertainty, average width (AW), is nearly 10%. This is extremely high, suggesting our source data is very imprecise. While this may be acceptable in the very early stages, we should try to reduce this source uncertainty before reaching a final design. Visually, this large uncertainty is represented by the large space between the *Plausibility* and *Belief* curves in Figure 37. Reducing uncertainty will shrink this gap and bring the curves closer together. Note the black **x** on the upper right portion of the BPA plot. This is our *Eval Point* (defined in Section 3.8) that we've used to represent the *PoF* of the system. It corresponds to 90% *Confidence* and 10% *Risk*.

Figure 38a shows the system's sensitivity to *PoF* uncertainty. Figure 38b shows the sensitivity to *PoF* reduction via improved parts and redundancy.



(a) (b)

Figure 38: Initial SPAARO ETFTA Sensitivity Analysis Results

7.3.1 Improvement 1: Onboard Engine Starter

The reliability sensitivity analysis results in Figure 38b reveal that it's most cost-effective to improve the engine. The PoF uncertainty sensitivity results confirm that there is quite a bit of PoF uncertainty associated with the engine, so this is the first thing we should improve. Students chose to include an onboard starter for the engine because one could argue that the starter is essentially the same thing as a redundant engine (without all of the integration challenges). The onboard starter is powered by a 12V LiPo battery and can start the engine in-flight after a failure.

Fuji-Imvac's 64-A 5.7HP gas engine with onboard starter was chosen for it's suitable power rating and starting capability. The engine, with starter, increases the UAV cost by $500 ($1000 total cost, $500 more than the original budgeted $500 without the starter). The two UAV pilots were surveyed again and the PoF ranges were updated as shown in Table 9.

Updating these values in the ETFTA code, the new reliability results are shown below:

```
PoF for the system: 0.1391028407
Failures:  13910.3 per 100,000 hours
Total System Cost: $3126
AW for the system: 0.068040
```

	Source 1	Source 2
Original Engine	{0.05, 0.1}	{0.01, 0.05}
Engine with Starter	{0.01, 0.03}	{0.005,0.01}
Net Cost: +$500		

Table 9: Updated Engine *PoF*

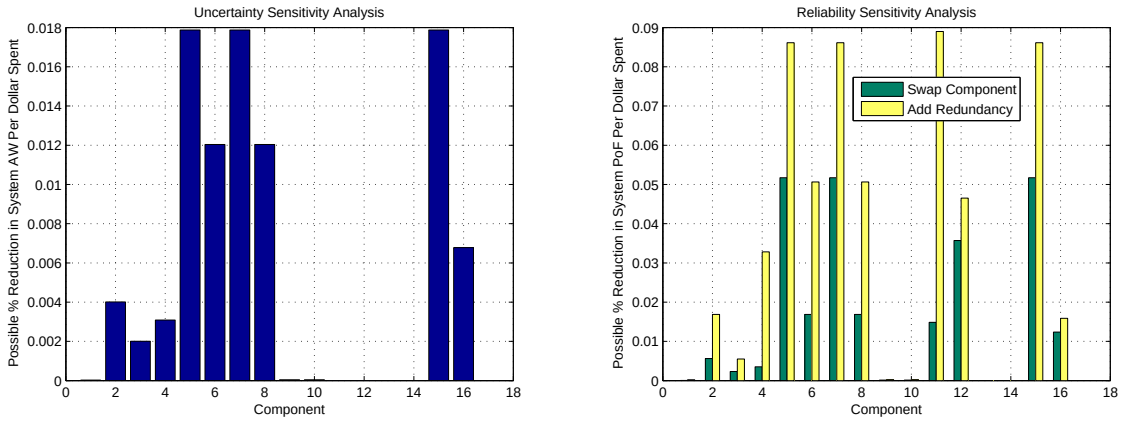
---------- Sensitivity Legend ----------

Part #	Description	Cost($)	MATI($)
1	72MHz Receiver	50	100
2	6V Battery	30	30
3	Signal Interference	1	1
4	Pilot Error	1600	5000
5	Rudder Servo	20	15
6	Rudder Surface	10	10
7	Elevator Servo	20	15
8	Elevator Surface	10	10
9	Left Aileron Servo	20	15
10	Right Aileron Servo	20	15
11	Nylon 1/4-20 Wing Bolt	5	15
12	Main Wing Spar - Carbon	300	200
13	Left Aileron Surface	10	10
14	Right Aileron Surface	10	10
15	Throttle Servo	20	15
16	Engine	1000	500

7.3.2 Improvement 2: Steel Wing Bolts

Figure 39b shows the reliability sensitivity analysis results for our next improvement. Note that component 16 (the engine) now has a reduced effect on system (un)reliability compared to the original engine without starter. From these results, it appears the nylon bolt is the

next most efficient improvement. Using steel bolts instead of nylon bolts increases weight but should increase reliability. Steel bolts have a tensile strength in excess of 36,000psi (compared to the nylon's 10,000psi). We could substitute this into the ETFTA toolbox to get an infinitesimally small PoF, but we can assume the UAV's structure will fail elsewhere before steel bolts are broken. We choose to ignore this part by setting PoF to zero. We do, however, account for the cost[12] of using these steel bolts.



(a) (b)

Figure 39: CASE STUDY: Improvement 1 Sensitivity Analysis Results

We've increased the system cost by $10 for these bolts and modified the ETFTA code[13].

7.3.3 Improvement 3: Futaba S3192 Servos

Recall from the sensitivity results from Figure 39b that components 5, 7, and 15 (servos) are the next best components to improve. As is evident from the uncertainty sensitivity analysis results in Figure 39a, these components also greatly affect the uncertainty in system PoF. Students researched servos and found Futaba S3192 digital, high-torque servos to be

[12]While steel bolts are cheap, they weigh much more than their nylon counterparts. We increase the "cost" of using these because this will increase our UAV's weight.

[13]While $10 was arbitrarily chosen, one could use a more accurate cost due to weight increase from researching SUAV costs and weights.

a better, more reliable (but more expensive) choice. An experiment was set up to test six of these servos until failure (Figure 40). Springs were used to simulate appropriate aerodynamic loads. The servos were run continuously at 0.85Hz until failures were recorded. Table 10 shows the *PoF* estimates for the original standard servo and updated estimates based on the experimental data.

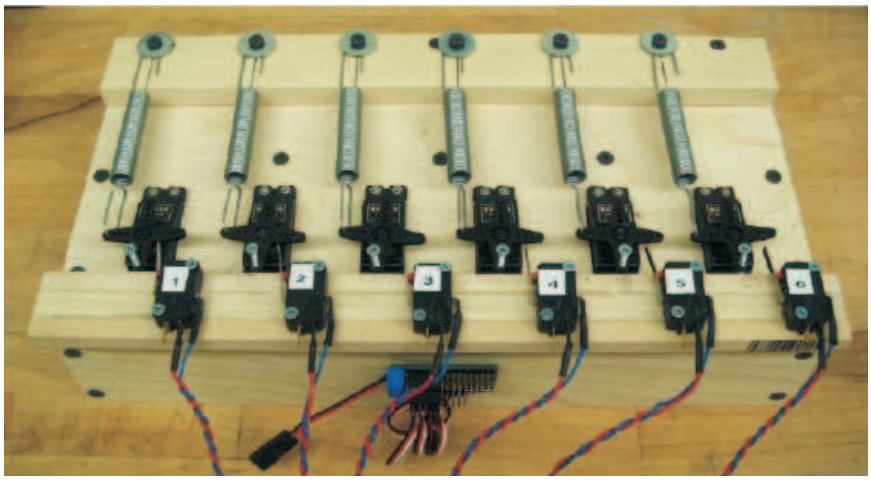


Figure 40: Servo Testing Rig

	PoF	Additional Cost
Original Servos	{0.002, 0.003}	–
Futaba S3192	{0.0013, 0.017}	$15 per servo

Table 10: Updated Servo *PoF*

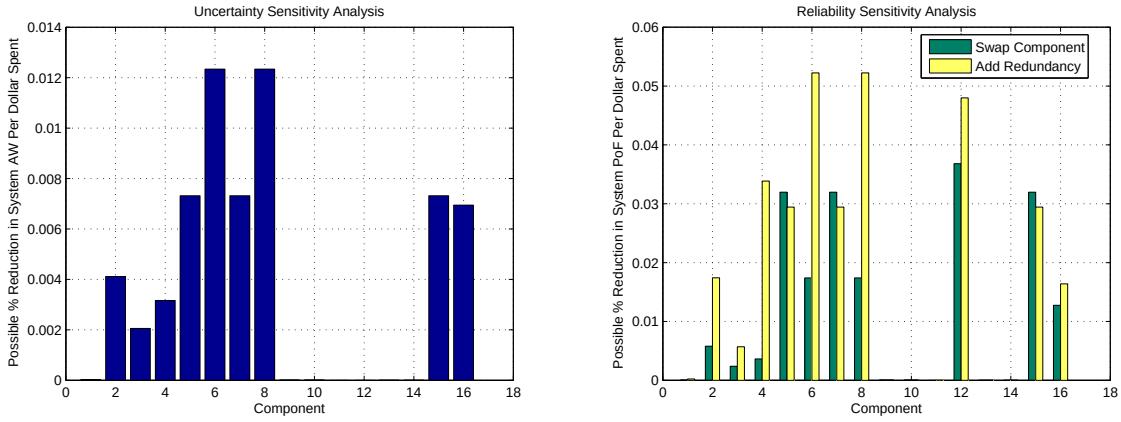
Updating the ETFTA code to reflect these new servos (for all servos on the UAV), we get the following results:

```
PoF for the system: 0.1354421550
Failures:  13544.2 per 100,000 hours
Total System Cost: $3211
AW for the system: 0.066642
```

---------- Sensitivity Legend ----------

Part #	Description	Cost($)	MATI($)
1	72MHz Receiver	50	100
2	6V Battery	30	30
3	Signal Interference	1	1
4	Pilot Error	1600	5000
5	Rudder Servo	35	15
6	Rudder Surface	10	10
7	Elevator Servo	35	15
8	Elevator Surface	10	10
9	Left Aileron Servo	35	15
10	Right Aileron Servo	35	15
11	Nylon 1/4-20 Wing Bolt	15	100
12	Main Wing Spar - Carbon	300	200
13	Left Aileron Surface	10	10
14	Right Aileron Surface	10	10
15	Throttle Servo	35	15
16	Engine	1000	500

Figure 41 shows the new sensitivity analysis results.



(a) (b)

Figure 41: CASE STUDY: Improvements 2 & 3 Sensitivity Analysis Results

7.3.4 Improvement 4: Redundant Control Surfaces and Servos

With the next most critical components being relatively similar in 5, 6, 7, 8, and 15 (servos and control surfaces), a major control overhaul is proposed – adding redundant surfaces and servos to all of these components. Conveniently, the basic configuration has a pusher propulsion system with twin tailbooms, so dual rudders is an easy upgrade. The elevator can be split in two to give two independent surfaces[14]. This design change warrants a change to the basic fault tree architecture. Figure 42 shows the new fault tree, and Table 11 shows the MCS breakdown with the new components. The ETFTA code was updated to reflect these changes, and Figure 43 shows the sensitivity analysis results.

It's obvious from Figure 43b that adding redundancy dramatically improves the reliability of the components. We see that components 5-15 now individually have very little effect on system (un)reliability because they are buried within redundant subsystems. Next, we see that the carbon wing spar is the next most critical improvement with "pilot error" also critically important. Adding an autopilot will greatly improve UAV mission repeatability and enable students to limit the load factor to 1.5g's – that is, the autopilot can be commanded to fly within a 1 to 1.5g vertical acceleration limit greatly reducing airframe stress. Load factor is a large contributor to structural reliability so adding an autopilot (reducing the load factor) will *also* reduce the PoF of the wing spar. Table 12 shows the updated spar loading conditions under this 1.5g limited load factor.

The wing spar loads are updated in the ETFTA inputs, as well as PoF data for the Piccolo II autopilot selected. The autopilot is paired with the "72MHz Receiver" in an AND gate because we have have two ways of controlling the UAV – *either* through the 900MHz autopilot link or directly through the 72MHz link. The "pilot error" component is switched to "operator error" to account for dangerous commands sent to the autopilot[15]. Additionally,

[14]When making this decision, the elevator sizing was increased by 20% to account for a single elevator failure. The UAV should be able to fly with only one elevator (or rudder, or aileron) so surfaces must be sized accordingly.

[15]This operator error could be greatly minimized or eliminated with a fully autonomous system where the

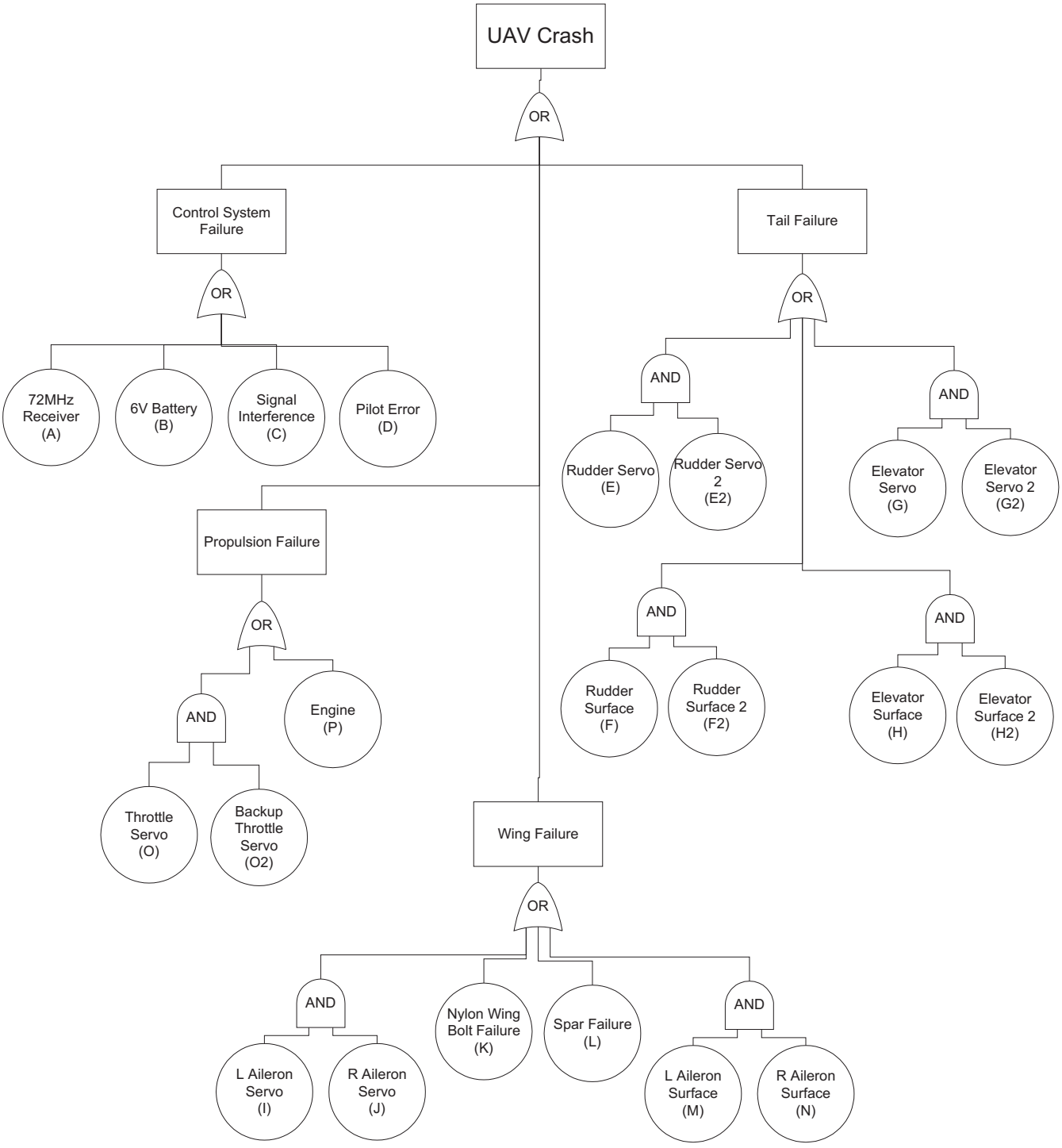
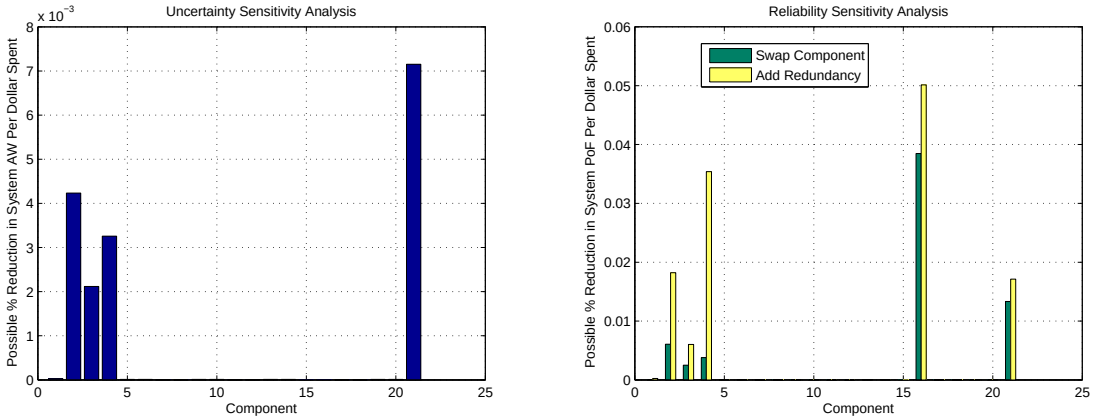


Figure 42: SPAARO Fault Tree: Redundant Control Surfaces (Improvement Step 4)

MCS	Label	Description	PoF Source 1	PoF Source 2	Cost ($)	$MATI$ ($)
1	A	72MHz Receiver	{1e-5, 2e-5}	{1e-5, 2e-5}	$50	$100
2	B	6V Battery	{2e-4, 1e-3}	{1e-4, 2e-4}	$30	$30
3	C	Signal Interference	{5e-6, 1e-5}	{1e-5, 2e-5}	$1	$1
4	D	Pilot Error	{0.03, 0.05}	{0.01, 0.1}	$1600	$5000
5	E	Rudder Servo 1	{1.3e-3, 1.7e-3}	{1.3e-3, 1.7e-3}	$35	$15
	E2	Rudder Servo 2	{1.3e-3, 1.7e-3}	{1.3e-3, 1.7e-3}	$35	$15
6	F	Rudder Surface	{2e-4, 1e-3}	{1e-4, 2e-4}	$10	$10
	F2	Rudder Surface	{2e-4, 1e-3}	{1e-4, 2e-4}	$10	$10
7	G	Elevator Servo 1	{1.3e-3, 1.7e-3}	{1.3e-3, 1.7e-3}	$35	$15
	G2	Elevator Servo 2	{1.3e-3, 1.7e-3}	{1.3e-3, 1.7e-3}	$35	$15
8	H	Elevator Surface 1	{2e-4, 1e-3}	{1e-4, 2e-4}	$10	$10
	H2	Elevator Surface 2	{2e-4, 1e-3}	{1e-4, 2e-4}	$10	$10
9	I	Left Aileron Servo	{1.3e-3, 1.7e-3}	{1.3e-3, 1.7e-3}	$35	$15
	J	Right Aileron Servo	{1.3e-3, 1.7e-3}	{1.3e-3, 1.7e-3}	$35	$15
10	K	Steel Wing Bolt	0	0	$15	$15
11	L	Main Wing Spar	0.0226	0.0226	$300	$200
12	M	Left Aileron Surface	{2e-4, 1e-3}	{1e-4, 2e-4}	$10	$10
	N	Right Aileron Surface	{2e-4, 1e-3}	{1e-4, 2e-4}	$10	$10
13	O	Throttle Servo 1	{1.3e-3, 1.7e-3}	{1.3e-3, 1.7e-3}	$35	$15
	O2	Throttle Servo 2	{1.3e-3, 1.7e-3}	{1.3e-3, 1.7e-3}	$35	$15
14	P	Engine	{0.05, 0.1}	{0.01, 0.05}	$1000	$500

Table 11: CASE STUDY: MCS Assignments for Updated Fault Tree



(a) (b)

Figure 43: CASE STUDY: Improvement 4 Sensitivity Analysis Results

Force	Mean Value, μ	Std Deviation, σ
S_{spar}(Bending Moment)	1261 lbs/in	100 lbs/in
R_{spar}(Bending Moment)	3600 lbs/in	500 lbs/in

Table 12: Updated Load Capacities for SPAARO's Carbon Wing Spar

an “Autopilot” component is added.

Below is the ETFTA output.

```
PoF for the system: 0.0278151729
Failures:  2781.5 per 100,000 hours
Total System Cost: $8336
AW for the system: 0.013626
```

---------- Sensitivity Legend ----------

Part #	Description	Cost($)	MATI($)
1	72MHz Receiver	50	100
2	6V Battery	30	30
3	Signal Interference	1	1
4	Operator Error	1600	2000
5	Rudder Servo 1	35	15
6	Rudder Servo 2	35	15
7	Rudder Surface 1	10	10
8	Rudder Surface 2	10	10
9	Elevator Servo 1	35	15
10	Elevator Servo 2	35	15
11	Elevator Surface 1	10	10
12	Elevator Surface 2	10	10
13	Left Aileron Servo	35	15
14	Right Aileron Servo	35	15
15	Nylon 1/4-20 Wing Bolt	15	100
16	Main Wing Spar - Carbon	300	200
17	Left Aileron Surface	10	10
18	Right Aileron Surface	10	10
19	Throttle Servo 1	35	15
20	Throttle Servo 2	35	15
21	Engine	1000	500
22	Piccolo II Autopilot	5000	2000

---

human is incapable of commanding dangerous maneuvers. But, tuning the autopilot to autonomously fly safely takes time which costs money

7.4 SPAARO Reliability Improvement Summary

This process of improving reliability in a step-by-step manner has been illustrated in the previous section. For conciseness, latter parts of the improvement steps have been omitted. Table 13 summarizes all reliability improvement steps considered (in the order suggested by the ETFTA sensitivity analyses).

Step	Improvement	Improvement Cost	Resulting PoF	System Cost
0	N/A (Initial Design)	N/A	0.1953	$2626
1	Onboard Engine Starter	$500	0.1391	$3126
2	Steel Wing Bolts	$10	0.1385	$3136
3	Futaba S3152 Servos	$75	0.1354	$3211
4	Redundant Surfaces and Servos	$125	0.1304	$3336
5	Piccolo II Autopilot	$5000	0.02709	$8336
6	Redundant 6V Battery	$30	0.02628	$8366
7	Engine Monitoring System	$1500	0.009289	$9866
8	Redundant Engine	$2500	0.00036887	$12366
9	Double Redundant Surfaces	$170	0.00036284	$12536
10	Operator Training[16]	$4800	0.00019975	$17336

Table 13: CASE STUDY: Possible SPAARO Reliability Improvements

While this list of improvements would dramatically improve SPAARO's reliability, the ballooning cost of such a system would make it impractical for academic development. To find the point of diminishing returns, system PoF was plotted vs. cost and is shown in Figure 44.

As we can see, the cost to increase reliability grows dramatically as improvements are made. Students decided that an engine health management system (step 7 in Table 13) would be the final improvement step for the SPAARO design. This improvement is highlighted by the asterisk on Figure 44. Indeed, this appears to be a good "elbow in the curve" where we

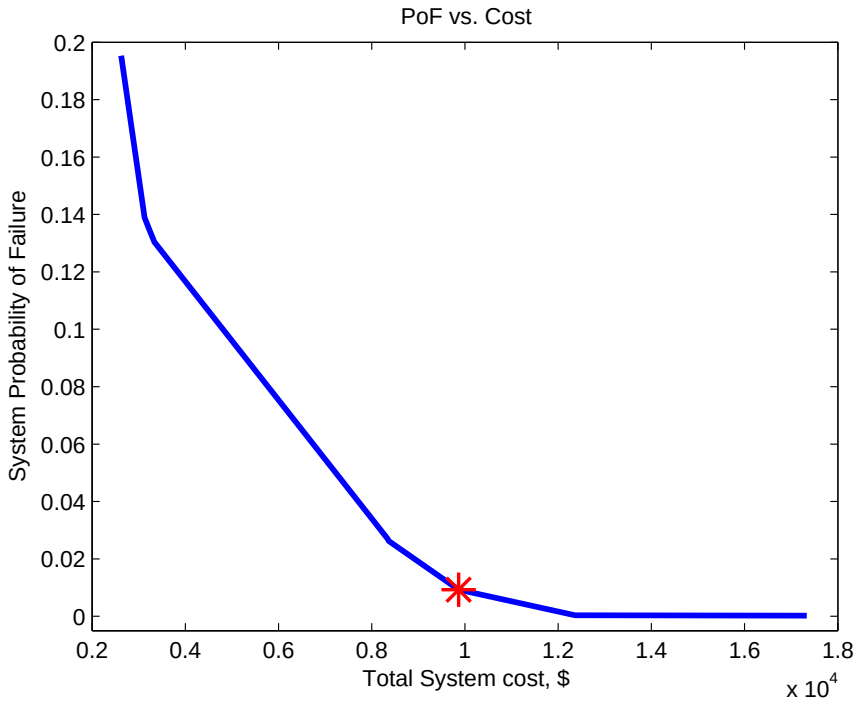


Figure 44: CASE STUDY: *PoF* vs. Cost for the SPAARO UAV

should stop spending money to improve reliability.

This process has improved the reliability of the SPAARO UAV design in a cost-effective manner. It is important to note that the sensitivity analysis results were used as a guide but not as rule – good engineering intuition was used as a "sanity check" for sensitivity analysis results before blindly making an improvement. Sometimes the locally cost-effective improvement was discounted for another improvement that made more practical sense. (E.g., adding the autopilot reduced the PoF of the wing spar, even though the wing spar was the highest ranked component on the sensitivity analysis.) These "sanity check" situations sprout from interdependence between parts. (For example, adding an autopilot improves the spar reliability by reducing flight loads so the autopilot and spar are related by some relationship the ETFTA toolbox may not account for, but an engineer can identify). A key assumption in this research is component independence which is overwhelmingly the case for small UAVs. Nevertheless, interdependence is an example of why a human engineer should always be in charge of design and not rely upon a black box to design the system. Often times sensitivity analyses do not account for all details a human designer is considering. The ETFTA toolbox, like any optimization routine or design tool, should be used to complement an engineer's design ability rather than replace it.

7.5 SPAARO Flight Testing

Three SPAARO UAVs were built by researchers in the NSL and subsequently flown in reserved military airspace at Fort Pickett in Blackstone, Virginia. Figure 45a shows the three assembled UAVs and Figure 45b shows a large payload (l0 lbs, 10x8x7 inches) mounted inside the forward payload bay.

The three SPAAROs were flown extensively for five days totalling 32 successful flights and 17 flight-hours. Figure 46a shows the SPAARO just after takeoff and Figure 46b shows a SPAARO flying overhead.

(a) (b)

Figure 45: SPAARO UAV Flight Testing Pictures 1 & 2

(a) (b)

Figure 46: SPAARO UAV Flight Testing Pictures 3 & 4

Over the course of the five days, experiments included four autonomous, synchronous flights of two UAVs, stability parameter identification doublets, specific excess power testing at various altitudes up to 6000 feet above ground level, glide tests for max lift to drag ratio, and autonomous "engine out" flight performance. The successful flight tests at Ft. Pickett validated the design and the SPAARO proved to be a reliable platform for a variety of experiments. The ETFTA Toolbox was a valuable tool in highlighting where improvements could be made to improve reliability while keeping the cost down.

8 Conclusions & Future Work

Fault Tree Analysis was proposed as a method to calculate a system's probability of failure. Using relatively simple boolean logic, AND and OR gates connect individual components to make up the system "tree" so analyses can be performed. Dempster-Shafer Theory (Evidence Theory) was suggested as a suitable way to organize imprecise and conflicting failure data that may be hard to obtain for many inexpensive parts. DST generalizes the data into sets of upper and lower bounds and BPA plots visually show a system's PoF ranges and uncertainty.

Three sensitivity analyses were proposed to highlight the most cost-effective components the UAV designer should improve. These sensitivity analyses lead the engineer through a sequence of cost-effective improvements to develop a more reliable UAV without the need for costly optimization or testing.

Structural reliability was discussed and two examples were given as to how to incorporate structural reliability into the design. Finally, a case study about the SPAARO design was presented to illustrate all methods discussed in this thesis.

The motivation for this thesis is derived from the fundamental problem that failure data for inexpensive, commercial-off-the-shelf parts is not readily available. Future work could focus on testing these smaller, inexpensive parts (where practical) that are increasingly being used in applications where reliability is of the utmost concern. While obtaining detailed failure data for specific COTS components will always be unlikely, future research could focus on testing "generic" or "representative" parts that could be used as "ballpark estimates" for similar parts – currently, there is *no* data readily available for many of these COTS parts. Further, basic construction techniques (i.e. hinging surfaces, vibration damping engine mounts, etc.) should be studied and standardized for future SUAV designs.

A key concept this thesis does not discuss is life-cycle cost. A system's true cost is not solely its upfront cost, discussed in this thesis, but its cost to operate over the product's entire

lifetime. PoF values can change as parts fatigue and the rates at which these parts fatigue is different for every part and application. Accounting for the various component fatigue rates should lead to an optimal "overhaul schedule" that would further improve the life-cycle cost-effectiveness of the system.

A Structural Reliability Example: Estimating Reliability of a Strut Braced Wing

The following is an illustrative process for designing a SUAV wing spar. Given certain geometric, performance, and reliability requirements, the spar should be as lightweight as possible. A baseline design is first evaluated for weight and reliability followed by three design changes: 1) varying the spar dimensions, 2) adding a simple strut and approximating bending moments, 3) adding a simple strut with a costly but accurate series of structural derivations to calculate bending moments.

Suppose we need to design a wing spar suitable for a new SUAV (20lbs) in the most cost and weight effective way. We'd also like the UAV to be extremely reliable, so the wing's failure rate must be less than 1 failure per 100,000 flight hours. Because this UAV is mostly a surveillance tool, we set the maximum load factor to positive 2.5g's; in other words, the wing structure should be able to support a $2.5 * 20$ lb $= 50$ lb load corresponding to turning flight, pull-up maneuvers, wind gusts, etc.

Due to cost concerns, hollow aluminum tube with a circular cross section will be used in lieu of carbon fiber alternatives which cost an order of magnitude more. The lift distribution is considered uniform as a simplifying, conservative assumption[17]. The outer diameter of the spar is also constrained to 0.75 inches because a thin airfoil was chosen. Other assumptions are given in Table 14.

Spar design consists of predicting the maximum stresses within the spar and verifying that the material strength is capable of supporting these stresses. Equation (49) is the basis for this stress calculation.

$$\sigma_{xx}(x, y) = \frac{-M(x) * y}{I_{xx}} \tag{49}$$

where M is the bending moment at any given span location (x), y is the vertical distance

[17]This is a close approximation to the elliptical distribution on high aspect-ratio wings. Further, calculated stresses at the wing root will be larger with this assumption than with a true elliptical distribution.

Table 14: Assumptions for Spar Design

Parameter	Value
Semi-span (b)	6 ft
Lift (per wing)	25 lbs
Lift Distribution	Uniform
Spar Material	Aluminum 2014-T6
ρ_{Al}	0.101 lbs/in^3
σ_{yield}	$\mu = 58,000$ psi, $\sigma = 5,000$ psi
Max AllowedPoF	$1 * 10^{-5}$

from the neutral axis (center of spar) to the maximum radius, and I_{xx} is the area moment of inertia of the cross section. Typically, the maximum stress is located at the wing root and at the vertical extremities of the spar[18]. Section A.1 discusses the "baseline" spar design and PoF evaluation. Section A.2 proposes using a thicker spar to reduce stresses at the root and increase reliability but this increases weight.

Another option to reduce stresses in the spar is to add a simple strut and use a slightly smaller main spar. This can be a lighter solution than adding a larger spar because it reduces the bending moment (and therefore stresses) at the root. Wing struts are generally statically indeterminant systems because stresses throughout the wing and strut are dependent on material properties and deflections. It can be difficult and time consuming to derive exact equations and solutions for these systems. Therefore, Section A.3 illustrates an estimation process where the wing-strut system is designed from two bounding bending moment estimations.

Finally, Section A.4 is the wing-strut evaluation after exhaustive system equation derivations.

[18]This applies to spars with constant cross section and elastic modulus, and a nearly elliptical lift distribution on the wings.

This section is presented to show that *informed* estimates from Section A.3 can be a more time(cost)-efficient way of finding structural properties than a time consuming (costly) high-fidelity system derivation.

A.1 Baseline Spar Design

Assuming a uniform lift distribution on the wing (Figure 47a), the bending moment at the root is easily calculated from the equivalent moment from a point load[19] (Figure 47b).

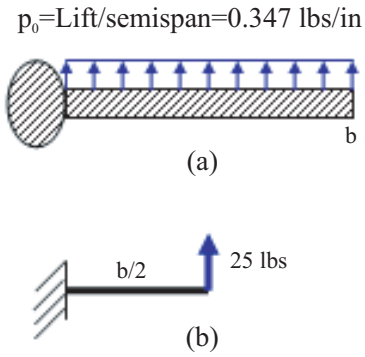


Figure 47: a) Uniform Lift Distribution on Wing, b) Equivalent Bending-Moment at Root

Here, $M_{\text{root}} = 25\text{lbs} * 36\text{in} = 900$ lb*in. The area moment of inertia of a circular spar is given by Equation (50):

$$I_{xx} = \frac{\pi}{4}(r_2^4 - r_1^4) \tag{50}$$

where r_2 and r_1 are the outer and inner radii of the circular spar, respectively. Choosing $r_2 = \frac{3}{8}$in and a wall thickness of $\frac{1}{16}$in gives $r_1 = \frac{5}{16}$in. The weight of each wing's spar is then:

$$W_{\text{spar}} = \pi * b * \rho_{\text{Al}}(r_2^2 - r_1^2) \tag{51}$$

Then, using Equations (49)-(51), $\sigma_{\max} = 41970$ psi and $W_{\text{spar}} = 0.9817$lbs. We set the stress standard deviation (σ) to 2000 psi to account for unpredicted stress variance.

[19]Note that in this section, b is the *semi-span* (half wingspan), not the full wingspan. In Section 7, b represented the *full* wingspan.

With this maximum expected bending stress and the yield stress from Table 14, we can calculate the spar's PoF with the procedure from Section 5. The resulting PoF is 0.001444, over two orders of magnitude higher than the requirement of $1 * 10^{-5}$. Table 15 lists the baseline spar results and Figure 48 shows graphical results for the PoF calculation.

Table 15: Baseline Spar Results

r_2	$\frac{3}{8}$in
r_1	$\frac{5}{16}$in
$W_{\text{each spar}}$	0.9817 lbs
$\sigma_{\max}$	41970 psi
PoF	**0.001444**
$W_{\textbf{total wing}}$	**1.963 lbs**

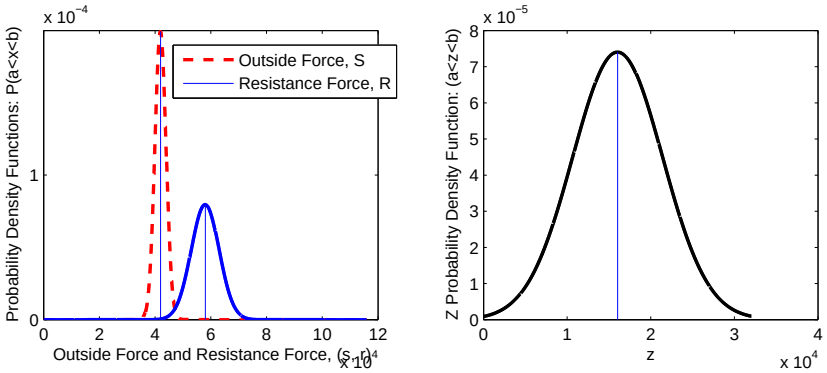


Figure 48: PoF Evaluation for Baseline Spar

A.2 Case 1: Increased Spar Size

Because the baseline spar does not meet the reliability requirement, the stress at the root must be decreased. With r_2 constrained to $\frac{3}{8}$in, r_1 can be decreased. In other words, we can increase the wall thickness of the spar to reduce the critical bending stress and reduce the probability of failure. However, this increases the spar weight.

The next largest available wall-thickness is $\frac{3}{32}$in. This gives $r_2 = \frac{3}{8}$in and $r_1 = \frac{9}{32}$in. Substituting these new values into Equations (49)-(51) we get $\sigma_{\max} = 31787$ psi and $W_{\text{spar}} = 1.405$ lbs – an increase of 0.85 lbs for both wings. The new PoF becomes $5.568 * 10^{-7}$ which satisfies the requirement. Table 16 summarizes Case 1 results.

Table 16: Case 1: Results

r_2	$\frac{3}{8}$in
r_1	$\frac{9}{32}$in
$W_{\text{each spar}}$	1.406 lbs
$\sigma_{\max}$	31787 psi
PoF	$5.568 * 10^{-7}$
$W_{\textbf{total wing}}$	**2.811 lbs**
Weight Increase from Baseline	**43%**

While increasing the wall thickness of the spar did reduce the PoF to a suitable level, it increased the total wing spar weight by 43%.

A.3 Case 2: Strut-Braced Wing with Estimated Bending Moments

Another option to reduce the stresses at the root is to add a strut to the wing to reduce the bending moment at the root. For this case, we choose to keep the original spar dimensions ($r_2 = \frac{3}{8}$in, $r_1 = \frac{5}{16}$in) and add a strut as shown in Figure 49. For simplicity, we choose the strut to be $\frac{1}{16}$in thick aluminum (2014-T6) with a chord of $\frac{1}{2}$in and a rectangular cross section[20].

[20]While we choose a simple rectangular cross section for this example problem, an airfoil shaped cross section will yield better buckling resistance and lower drag.

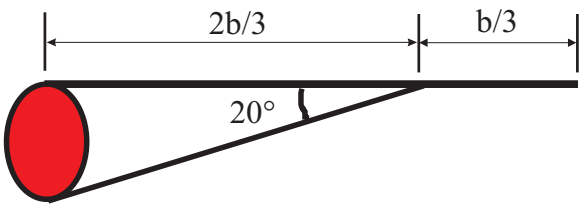


Figure 49: Strut-Braced Wing Geometry

We'd like to estimate the reliability of this system without deriving the complex system equations. A strut-braced wing in this form is a statically-indeterminant system that requires knowledge of deflections and material stiffness properties. For this section, we assume we simply don't have the human resources to solve such a system.

From this geometry, we estimate the strut will relieve one-third of the root bending moment. Recall that this moment for the plain wing (Section A.1) is 900 lb*in, so we estimate the new bending moment to be 600 lb*in, ±10%. This corresponds to estimated bending moment bounds of $[510, 690]$ lb*in.

Using the methods in Section 5, we calculate the following results for this case (Table 17). Note the bounded σ and PoF values.

Table 17: Case 2: Results

r_2	$\frac{3}{8}$in
r_1	$\frac{5}{16}$in
$W_{\text{each spar}}$	0.9817 lbs
$W_{\text{each strut}}$	0.1568 lbs
$\sigma_{\max}$	[23783, 32177] psi
PoF	$[1.031 * 10^{-10}, 8.013 * 10^{-}7]$
$W_{\textbf{total wing}}$	**2.278 lbs**
Weight Increase from Baseline	**16%**

We can see that adding the strut has increased the spar's reliability without adding near as much weight as case 1 (thicker spar). In fact, we save almost 30% weight savings when using the strut instead of the thicker spar.

A.4 Case 3: Strut-Braced Wing with Exact Bending Moments

While the previous section concludes that adding the strut will increase reliability in a more weight-effective way than using a thicker spar, the quantitative results are dependent on the initial assumption that the strut will reduce the root bending moment by one-third. This assumption allowed us to calculate the system PoF bounds in a fairly quick and painless process. To illustrate the ease and effectiveness of the previous estimation when compared to the exact answer, the complex system equations were derived[21] for this system and are shown in Equations (52)-(58).

$$F = \frac{-T_x p_0}{2E_w I_w}\left[\frac{T_x b}{3} - \frac{T_x^2}{12} - \frac{b^2}{2}\right]\left[\frac{\tan\theta}{E_s A_s} + \frac{T_x^2 \sin\theta}{E_w I_w}\right]^{-1} \tag{52}$$

$$\Delta_{\text{strut}} = \frac{F T_x}{E_s A_s \cos(\theta)} \tag{53}$$

$$\Delta_{W_{T_x}} = \Delta_{\text{strut}} \sin(\theta) \tag{54}$$

$$V_{\text{inboard}} = p_0 b - F\sin(\theta) - p_0 x \tag{55}$$

$$V_{\text{outboard}} = p_0 b - p_0 x \tag{56}$$

[21] While this derivation is good representation of the actual forces in the wing and strut (Figure 49), they are derived assuming frictionless pins on the strut-wing and fuselage connections, a uniform lift distribution, and negligible axial forces in the wing.

$$M_{\text{inboard}} = p_0 bx - F\sin(\theta)x - \frac{p_0 x^2}{2} + F\sin(\theta)T_x - \frac{p_0 b^2}{2} \tag{57}$$

$$M_{\text{outboard}} = p_0 bx - \frac{p_0 x^2}{2} - \frac{p_0 b^2}{2} \tag{58}$$

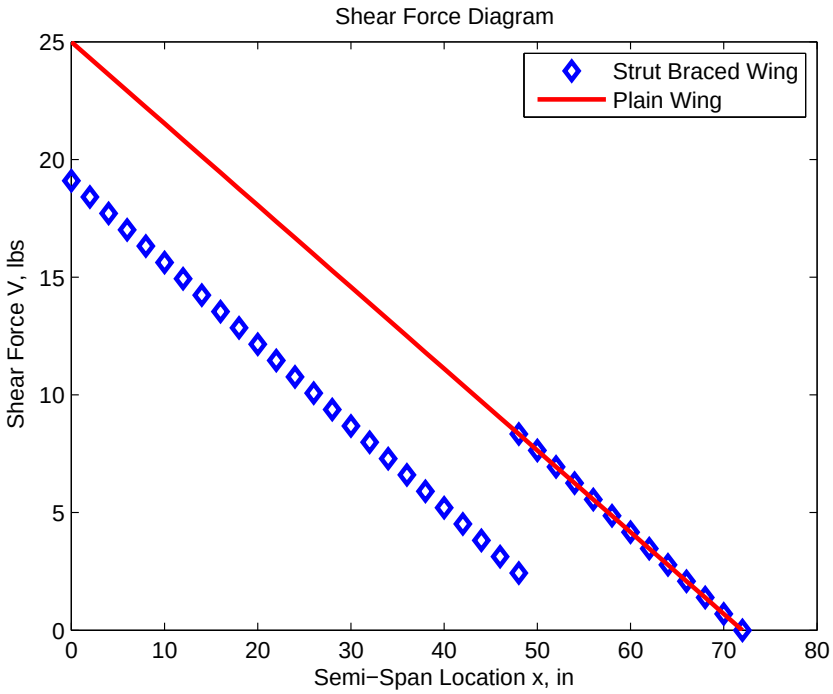
Equations (52)-(53) yield the (tensile) force in the strut and elongation of the strut, respectively. Equation (54) gives the vertical displacement of the wing-strut pin joint, and Equations (55)-(56) give the shear force distributions inboard and outboard of the joint, respectively[22]. Finally, Equations (57)-(58) give the moment distributions inboard and outboard of the joint, respectively. To solve the system of equations, we substitute the values given in Table 18.

Table 18: Case 3: System Equation Inputs

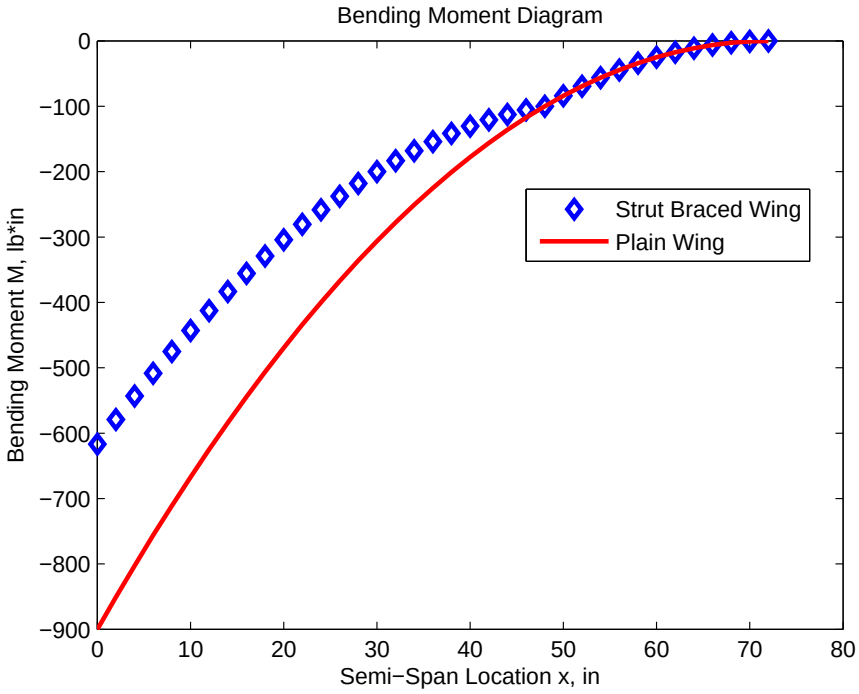
Description	Variable Designation	Value	Units
Distributed Lift Force	p_0	0.3472	lbs/in
Spanwise location of joint	T_x	48	in
Wing Modulus of Elasticity	E_w	$10.2 * 10^6$	lbs/in^2
Wing Area Moment of Inertia (I_{xx})	I_w	0.00804	in^4
Wing Semi-Span Length	b	72	in
Strut Angle	θ	20	degrees
Strut Modulus of Elasticity	E_s	$10.2 * 10^6$	lbs/in^2
Cross Sectional Area of Strut	A_s	0.03125	in^2

From these input values (also used in Sections A.1-A.3, where appropriate) MATLAB® was used to generate shear force and bending moment diagrams for the plain wing and strut braced wing. Figures 50a and 50b show the diagrams for both configurations.

[22] x is always defined as the distance from root to any given spanwise location.



(a)



(b)

Figure 50: Shear Force and Bending Moment Diagrams for Baseline and Case 3

As expected, the diagrams are identical outboard of the strut joint. Inboard of the joint, the strut alleviates a uniform 5.9 lbs of shear force and an increasing amount of bending moment towards the root. At the root, the bending moment decreases from 900 lb*in from the plain wing to 617 lb*in for the strut braced wing. Further, we find there is 17.3 lbs of tensile force in the strut and the wing deflects only 0.0028 inches vertically at the strut-wing joint (compared to 8 inches at the same location on the un-braced wing).

With an accurate bending moment (617 lb*in), we again calculate the PoF of this structure. (Results are shown in Table 19.)

Table 19: Case 3: Results

r_2	$\frac{3}{8}$in
r_1	$\frac{5}{16}$in
$W_{\text{each spar}}$	0.9817 lbs
$W_{\text{each strut}}$	0.1568 lbs
$\sigma_{\max}$	28759 psi
PoF	$2.775 * 10^{-8}$
$W_{\text{total wing}}$	2.278 lbs
Weight Increase from Baseline	16%

We see that the assumption from Case 2 (the bending moment will be one-third less with the strut) is accurate; the high-fidelity PoF value from Case 3 is bounded by the estimates from Case 2. Our "quick and dirty" method to find the PoF based on an educated guess resulted in a fairly accurate representation of the system's high-fidelity, exhaustive PoF calculation. While it is somewhat dangerous to intentionally forgo high-fidelity modelling and prediction, this section suggests that we can use bounded force ranges to approximate failure probabilities for complex structures and systems where high-fidelity modelling is not practical or possible.

B SPAARO Case Study ETFTA Toolbox Inputs (non-abbreviated)

```
% ———————————— DEFINE EVALUATION PARAMETERS ————————————————
% Define Confidence, Risk, and Amount of Improvement:
% ————————————————————————
% Confidence level:  What percent of evidence do you want to represent the
% model?  90%?  95%?  100%?   (enter 0 to 1)
Confidence=0.90;

% Risk:  How much do you want to trust the Plausibility function?  0
% ignores plausibility, represents the model ONLY from Belief function (most
% conservative).  1 represents the model from the Plausibility function (most
% aggressive, risky assumption)  (enter 0 to 1)
Risk=.20;

% PercentUncertReduction:  Assuming more research is done on a component,
% how much can you assume the uncertainty will be improved by?
% (enter 0 to 100)
PercentUncertReduction=20;

% PercentPOFReduction:  Assuming we swap a component for a higher quality
% alternative, how much can you assume the PoF will be improved by?
% (enter 0 to 100)
PercentPOFReduction=50;

% ————————————— INPUT COMPONENT DATA ———————————————————

% component A evidence:
JustinA=[1e-05, 2e-05,1];  % DSV input.  Model: [PoF_low, PoF_high, weight]
% component A descriptions:
```

```
A.dsv=JustinA;
A.cost=50;   % Upfront cost
A.description='72MHz Receiver';
A.MATI=100;  % Money Allocated to Improve

% component B evidence:
JustinB=[2e-04, 1e-03,1];
CraigB=[1e-04, 2e-04,1];
% component B descriptions:
B.dsv=combine([JustinB,CraigB]);  %combines two DSV ranges into a joint DSV
B.cost=30;
B.description='6V Battery';
B.MATI=30;

JustinC=[5e-06,1e-05,1];
CraigC=[1e-05, 2e-05,1];
C.dsv=combine([JustinC,CraigC]);
C.cost=1;
C.description='Signal Interference';
C.MATI=1;

JustinD=[0.03, 0.05,1];
CraigD=[0.01, 0.1,1];
D.dsv=combine([JustinD,CraigD]);
D.cost=1600;  % Based on 80 hours of training at 20 dollar/hour wage
D.description='Pilot Error';
D.MATI=5000;  % Money allocated for an autopilot

JustinE=[2e-03, 3e-03,1];
%JustinE=[0, 0,1];
E.dsv=combine([JustinE]);
E.cost=20;
E.description='Rudder Servo';
E.MATI=15;
```

```
JustinF=[2e-04, 1e-03, 1];
CraigF=[1e-04, 2e-04,1];
F.dsv=combine([JustinF,CraigF]);
F.cost=10;
F.description='Rudder Surface';
F.MATI=10;

JustinG=[2e-03, 3e-03,1];
G.dsv=combine([JustinG]);
G.cost=20;
G.description='Elevator Servo';
G.MATI=15;

JustinH=[2e-04, 1e-03,1];
CraigH=[1e-04, 2e-04,1];
H.dsv=combine([JustinH,CraigH]);
H.cost=10;
H.description='Elevator Surface';
H.MATI=10;

JustinI=[2e-03, 3e-03,1];
I.dsv=combine([JustinI]);
I.cost=20;
I.description='Left Aileron Servo';
I.MATI=15;

JustinJ=[2e-03, 3e-03,1];
J.dsv=combine([JustinJ]);
J.cost=20;
J.description='Right Aileron Servo';
J.MATI=15;

% S is the outside force, R is the resistance (strength) of the structure
```

```
%XXX=structurePoFcalc([muSLow, sigmaSLow; muSHigh, sigmaSHigh],...
% [muRLow, sigmaRLow, muRHigh, sigmaRHigh]);
JustinBoltForces=structurePoFcalc([165, 20; 165, 20],[490, 100; 490, 100]);
K.dsv=JustinBoltForces;
K.description='Nylon 1/4-20 Wing Bolt';
K.cost=5;
K.MATI=15;

SparForces=structurePoFcalc([2521, 200; 2521, 200],[3600, 500; 3600, 500]);
L.dsv=SparForces;
L.description='Main Wing Spar - Carbon';
L.cost=300;
L.MATI=200;

JustinM=[2e-04, 1e-03,1];
CraigM=[1e-04, 2e-04,1];
M.dsv=combine([JustinM,CraigM]);
M.cost=10;
M.description='Left Aileron Surface';
M.MATI=10;

JustinN=[2e-04, 1e-03,1];
CraigN=[1e-04, 2e-04,1];
N.dsv=combine([JustinN,CraigN]);
N.cost=10;
N.description='Right Aileron Surface';
N.MATI=10;

JustinO=[2e-03, 3e-03,1];
O.dsv=combine([JustinO]);
O.cost=20;
O.description='Throttle Servo';
O.MATI=15;
```

```
JustinP=[5e-02, 1e-01,1];
CraigP=[1e-02, 5e-02,1];
P.dsv=combine([JustinP,CraigP]);
P.cost=500;
P.description='Engine';
P.MATI=500;

% ------------ INPUT MCS STRUCTURE -------------------------
MCS(1).component{1}=A;
MCS(2).component{1}=B;
MCS(3).component{1}=C;
MCS(4).component{1}=D;
MCS(5).component{1}=E;
MCS(6).component{1}=F;
MCS(7).component{1}=G;
MCS(8).component{1}=H;
MCS(9).component{1}=I;  % components I & J are related by an AND gate.
MCS(9).component{2}=J;  % components I & J are related by an AND gate.
MCS(10).component{1}=K;
MCS(11).component{1}=L;
MCS(12).component{1}=M; % M & N are related by an AND gate.
MCS(12).component{2}=N; % M & N are related by an AND gate.
MCS(13).component{1}=O;
MCS(14).component{1}=P;
% ------------- END OF INPUTS --------------------

% ------------- RUN COMMANDS --------------------
% Condition DSV entries so Plaus<Bel
MCS=conditionDSVs(MCS);

% Run the system model
SysModel=systemDSV(MCS);

% Calculate "System PoF" of the Original system:
```

```
PoFSystem=systemPoF(SysModel);
fprintf('PoF for the system: %0.10f\n',PoFSystem)
fprintf('Failures:  %0.1f per 100,000 hours \n ',PoFSystem*100000);

% Calculate total cost of the system
TotalSystemCost=systemCost(MCS);
fprintf('Total System Cost: %0.0f\n',TotalSystemCost)

% Calculate "Average Width" of the Original system:
AWsystem=AW(SysModel);
fprintf('AW for the system: %f\n\n\n',AWsystem)

% Plot the new system BPA
plotBPA(SysModel, PoFSystem)

% Run uncertainty sensitivity analysis.  This generates a bar graph showing
% possible percent reduction in system AW (per dollar spent), if the respective
% evidence is reduced by an amount "PercentUncertReduction".
uncertSense(SysModel,MCS,PercentUncertReduction);

% Run PoF sensitivity analysis.  This generates a bar graph showing
% possible percent reduction in system PoF (per dollar spent), if the respective
% component's PoF is reduced by "PercentPOFReduction" OR .
PoFSense(SysModel,MCS,PercentPOFReduction);

PrintLegend(MCS)
```

References

[1] Mazen Ba-abbad, Efstratios Nikolaidis, and Rakesh Kapania. New approach for system reliability-based design optimization. *AIAA Journal*, 44, No. 5, 2006.

[2] Richard E. Barlow. *Engineering Reliability*. SIAM, 1998.

[3] S. Cook. NAVAIR personal communication. 2008.

[4] M.C. Cotting, J.F. Murtha, L. Techy, and C.A. Woolsey. Examples of augmentation of an atmospheric flight mechanics curriculum with an unmanned aerial vehicle. In 27^{th} *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Chicago, Illinois, August 10-13 2009 (To Appear).

[5] Arthur P. Dempster. A generalization of Bayesian inference. *Journal of the Royal Statistical Society*, Series B, vol. 30:205–247, 1968.

[6] M. Dermentzoudis. Establishment of models and data tracking for small UAV reliability. Master's thesis, Naval Postgraduate School, Monterey, CA, 2004.

[7] D.H. Dumas. Six facets of the open COTS box. In *RTO IST Symposium on "Commercial Off-The-Shelf Products in Defence Applications "The Ruthless Pursuit of COTS""*, volume MP-048, 2000.

[8] C.A. Ericson. Fault tree analysis - a history. In *Proceedings of The 17th International System Safety Conference*, 1999.

[9] Nasser S. Fard. Determination of minimal cut sets of a complex fault tree. volume 33, Nos 1-2, pages 59–62, 1997.

[10] Scott Ferson and W. Troy Tucker. *Sensitivity in risk analyses with uncertain numbers*, 2006.

[11] D. Stirzaker G. Grimmett. *Probability and Random Processes*. Oxford University Press Inc, third edition, 2001.

[12] A.A. Giunta, O. Golividov, D.L. Knill, B. Grossman, R.T. Haftka, W.H. Mason, and L.T. Watson. Multidisciplinary design optimization of advanced aircraft configurations. In *Fifteenth International Conference on Numerical Methods in Fluid Dynamics*, volume 490, pages 14–34, Springer-Verlag, Berlin, 1997.

[13] I. Hacking. Jacques Bernoulli's art of conjecturing. *The British Journal for the Philosophy of Science*, 22:209–229,, 1971.

[14] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[15] C. Kara-Zaitri. An improved minimal cut set algorithm. volume Vol. 13, No. 2, pages 114–132, 1996.

[16] J. M. Keynes. *A Treatise on Probability.* Macmillan, 1921.

[17] R. Kohl. Determining the suitability of cots for mission critical applications. In *RTO IST Symposium on "Commercial Off-The-Shelf Products in Defence Applications "The Ruthless Pursuit of COTS""*, 2000.

[18] P. Limbourg. *IPP Toolbox for MATLAB.* http://www.uni-duisburg-essen.de/il/ipptoolbox.php, 2007.

[19] P. Limbourg, R. Savic, J. Petersen, and H. Kochs. Fault tree analysis in an early design state using the Dempster-Shafer theory of evidence. *Risk, Reliability and Societal Safety - Aven and Vinnem*, 2007.

[20] R. E. Melchers. *Structural Reliability.* Ellis Horwood Limited, 1987.

[21] N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44:335–341, 1949.

[22] J.F. Murtha, M.C. Cotting, L. Techy, and C.A. Woolsey. The educational impact of designing, building, and testing a new UAV for curriculum enhancement. In $27^{th} AIAA$

Atmospheric Flight Mechanics Conference and Exhibit, Chicago, Illinois, August 10-13 2009 (To Appear).

[23] W. Oberkampf. Uncertainty quantification using Evidence Theory. 2005.

[24] Office of Secretary of Defense. *Unmanned Aircraft Systems Roadmap 2005-2030.*

[25] Office of Secretary of Defense. *Military Standard: Procedures for Performing A Failure Mode Effects and Criticality Analysis*, 1980. MIL-STD-1629A.

[26] R. Peoples and K. Willcox. A value-based MDO approach to assess business risk for commercial aircraft design. In *Journal of Aircraft*, volume 43.

[27] N.H. Roberts, W.E. Vesely, D.F. Haasl, and F.F. Goldberg. *Fault Tree Handbook.* Washington, D.C., 1981.

[28] N.F. Schneidewind. The ruthless pursuit of the truth about COTS. In *RTO IST Symposium on "Commercial Off-The-Shelf Products in Defence Applications "The Ruthless Pursuit of COTS""*, volume MP-048, 2000.

[29] K. Sentz and S. Ferson. Combination of evidence in dempster-shafer theory. Technical report, Sandia National Laboratory Tech Report, 2002. SAND 2002-0835.

[30] G. Shafer. *A Mathematical Theory of Evidence.* Princeton University Press, 1976.

[31] J. Sobieszczanski-Sobieski and R.T. Haftka. Multidisciplinary aerospace design optimization - survey of revent developments. In *AIAA 34th Aerospace Sciences Meeting and Exhibit*, volume 1996-0711.

[32] W. Zhao, T. Fang, and Y. Jiang. *Data Fusion using improved Dempster-Shafer Evidence Theory for Vehicle Detection*, 2002.