UNCERTAINTY QUANTIFICATION WITH MITIGATION ACTIONS FOR AIRCRAFT CONCEPTUAL DESIGN

A Thesis
Presented to
The Academic Faculty

by

Joseph Scott Wilson

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
May 2015

UNCERTAINTY QUANTIFICATION WITH MITIGATION ACTIONS FOR AIRCRAFT CONCEPTUAL DESIGN

Approved by:

Professor Dimitri N. Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Jimmy Tai
School of Aerospace Engineering
Georgia Institute of Technology

Professor Brian German
School of Aerospace Engineering
Georgia Institute of Technology

Professor Sankaran Mahadevan
School of Civil and Environmental Engineering
Vanderbilt University

Dr. Taewoo Nam
Frontier Research Development Department
Toyota Research Institute of North America

Date Approved: 03 April 2015

ACKNOWLEDGEMENTS

This has been a long journey, and I never could have made it this far without the immense help I was given along the way. I would like to thank the members of my committee. First, I need to thank my advisor, Professor Dimitri Mavris, who was instrumental in enabling me to compete this work. Also, I must thank him for giving me the time and resources to pursue this crazy course of action. I must thank Dr. Jimmy Tai for his suggestions for his advice, patience, and the roughly one hundred hours of meetings, even if we spent half of the time arguing. I want to thank Professor Brian German for all his help with aircraft design, hypotheses, engine design, and for showing me how a schlieren photograph could lie. Professor Sankaran Mahadevan has been a great help on the working with the reliability-based problems and has always been a pleasure to work with. I cannot forget Dr. Taewoo Nam for helping me define my ideas before the proposal and again after the defense as well as all the times we worked together while he was still at Georgia Tech.

I must express my gratitude to all the other people who helped me work out parts of this problem I wish to thank Roger Cooke, Sankaran Mahadevan, and Thomas Zang for helping me when I was developing the initial formulation of my methodology. I also want to thank the Daniel Cooksey, Elizabeth Saltmarsh, Kelly Griendling, Nick Molino, John Salmon, Daniel Garmendia, Brad Robertson, and all the others who attended my practice presentations and helped me refine my story.

I would be very remiss if I did not thank Dr. Michelle Kirby for taking a chance on me back when I was an undergrad with absolutely zero skills, putting up with my attitude, and encouraging me through all the trials between then and now. I want to thank all the people who have helped me with aircraft and engine modeling over the years. This includes but is not limited to Jeff Schutte, Chris Perullo, Donwook Lim, Taewoo Nam, Paul Brett, Jon Gladin, Jon Sands, Brian Kestner, and many others whose names are escaping me at the moment. Finally, I have to thank the Elizabeth Saltmarsh, Elizabeth Wilson, Joe Wilson, Martha Wilson, Paul Brett, Janet Trelease, and Daniel Cooksey for their help editing a much larger and more confusingly-worded document than should have been required.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The aircraft design process is characterized by increasing certainty and decreasing design freedom over time. This means that some of the most important design decisions are made at a time when little information is available to quantify the impact of those decisions. The goal of most advanced design methods is to increase the information available at the conceptual design stage to allow the designer to make the best decision possible. However, by the nature of the problem, there will always be some differences between the performance estimates computed during conceptual design and the performance of the final product. These differences may result in performance constraint violations, which can have severe financial impacts. As a result, constraint violations may necessitate downstream design changes to bring the aircraft back into compliance with requirements; these design changes will also have impacts on both cost and schedule. The ability to estimate the likelihood of late-stage design changes and the impact of said changes is key to mitigating the overall risk of a design and ensuring the business success of a product.

Reliability methods already exist to account for design uncertainty, and they have been applied to aircraft conceptual design studies. These techniques measure the likelihood that a design will comply with constraints by assessing the aircraft's performance under simulated uncertainty. However, existing reliability methods are not formulated to easily account for some important aspects of aircraft design. In fact, such methods are unable to account for much of the complexity associated with the sizing process commonly employed during conceptual design. Few methods account for the staged nature of aircraft design, and those that do often ignore some aspects

of the stages of design; specifically, reliability methods rarely deal with the concept of the design freeze associated with the transition from aircraft conceptual design to later design stages. Reliability methods also fail to account for the remedial nature of design when a performance constraint is violated due to uncertainty.

These remedial elements mitigate undesirable performance outcomes resulting from the development of uncertain design parameters throughout the design process. These "mitigation actions" bring the design back into compliance with constraints by adjusting design parameters external to the initial set of conceptual design parameters. By accounting for these mitigation actions, new reliability metrics can be developed. In addition to capturing the probability of compliance of the design, the designer can determine the probability of success after accounting for outcomes which can be recovered through mitigation actions. This additional information helps to determine the true likelihood that a design can successfully meet all of the requirements, giving the conceptual designer access to additional information which will enable better design decision making.

This work describes the development of a method by which the gaps identified within existing reliability methods can be filled. Hypotheses are developed in an attempt to fill the gaps which exist between reliability methods found in literature and the aircraft design process. Filling these gaps results in the development of a design methodology referred to as Aircraft Recovery through Mitigation & Optimization under Uncertainty for Reliability (ARMOUR). The ARMOUR method leverages traditional Reliability-Based Design Optimization and augments it to include concepts specifically taken from aircraft design including aircraft sizing, uncertainty margins, and mitigation actions. Within this thesis, a step-by-step formulation of the ARMOUR methodology is created which describes the necessary phases from initial identification of the need for an aircraft though model development and execution of the method. The step-by-step formulation is demonstrated on the conceptual design of a large civil transport aircraft. The ARMOUR method is then tested by comparing the new capabilities it enables to previously existing aircraft uncertainty assessment methods. As part of this process, the detriments of potential alternate implementations are demonstrated. Once tested and constructed, ARMOUR is exercised to explore relationships which were obscured or unobservable without it.

The contribution to the field of probabilistic aircraft conceptual design is the concurrent quantification of three elements in one decision making environment: the probability of compliance (from traditional reliability), the total probability of design recovery after failure (by modeling mitigation actions), and traditional design criteria such as vehicle weight or block fuel (from sizing). This new information affords the decision maker the ability to query the trade space between a design's overall probability of success[1] and traditional design criteria, creating a novel way to quantify the trade-off between risk and performance and to target a specific level of compromise between the two. The ARMOUR methodology enables the selection of design variables and uncertainty margins which both meet reliability goals for the compliance and success of the aircraft and also simultaneously optimize a traditional aircraft design performance metric. In this way, a design that efficiently meets reliability requirements can be found.

[1]Throughout this thesis, the overall probability of success means the design's probability of success after accounting for potential mitigation actions.

CHAPTER I

INTRODUCTION

The commercial aircraft design process is characterized as a long and expensive endeavor that balances high risk with high rewards. The design of a new aircraft can take many years from concept to entry into fleet and can cost billions of dollars [68]. The potential payout of a successful aircraft design can be significant. The guiding motivation for this thesis is to improve civil aircraft conceptual design by bringing information from preliminary design forward into the conceptual design stage. Specifically, there is a desire to quantify the impact of design uncertainty on aircraft performance as well as the total likelihood that a design will meet performance goals. This information must be processed such that it will be useful to conceptual designers, allowing them to select designs with some advanced knowledge of the potential downstream impacts of their choices.

To accomplish this goal, the design process of a civil transport aircraft is examined as it relates to the stages of design. How and why parameter uncertainty will vary for progressive stages is examined. The impact of uncertainty on aircraft performance is established. Also, a theory for how a design team could react to performance shortfalls resulting from uncertainty realizations is developed and a template for constructing representative actions is proposed. These effects are integrated into a methodology which will allow for the assessment of multiple reliability metrics. One metric is the probability of compliance: a reliability measure which is based on traditional uncertainty assessment methods. The second metric is the probability of recovery, a new metric that indicates the likelihood that a designed aircraft can be "recovered" through actions available to design teams in later stages of aircraft development.

The resulting overall methodology integrates this information to allow the designer to select a design concept with more information than would otherwise be available during conceptual design.

1.1 Aircraft Design Uncertainty

The decisions made during the conceptual design stage of the civil aircraft design process are often the most impactful on the final performance and cost of a product. Figure 1 illustrates this concept by demonstrating conceptually how changes occur during the design process, focusing on the advantages of Integrated Product and Process Development (IPPD) [82]. IPPD specifics are outside the scope of this thesis, but the principles shown in the diagram are still useful. The stages of design are illustrated as proceeding from left to right on the diagram. The number of changes for a traditional 'Serial' approach to design within each stage of the design process is compared with the number of changes associated with IPPD. IPPD focuses on more changes earlier on in the design process. The reason for this is illustrated by the third line, "Cost of Change." This curve is meant to show qualitatively that changes which occur further along in a design process will become increasingly expensive. Just as with IPPD, this thesis is intended to help designers make the best early decisions possible, thus avoiding heavy costs down the line.

Motivated by the cost of changes later in design and emerging techniques, a paradigm shift was predicted in Aerospace Engineering by the National Science Foundation's 1996 Strategic Planning Workshop [83]. Figure 2 illustrates what this paradigm shift would look like conceptually. This diagram shows "Today's Design Process" compared to a theoretical "Future Design Process," and attempts to show the advantages therein. Again, the x-axis illustrates the stages of design, proceeding from left to right. The primary goals of this paradigm shift are threefold. First is to delay any commitment to the cost of a design until later, illustrated by the "Cost

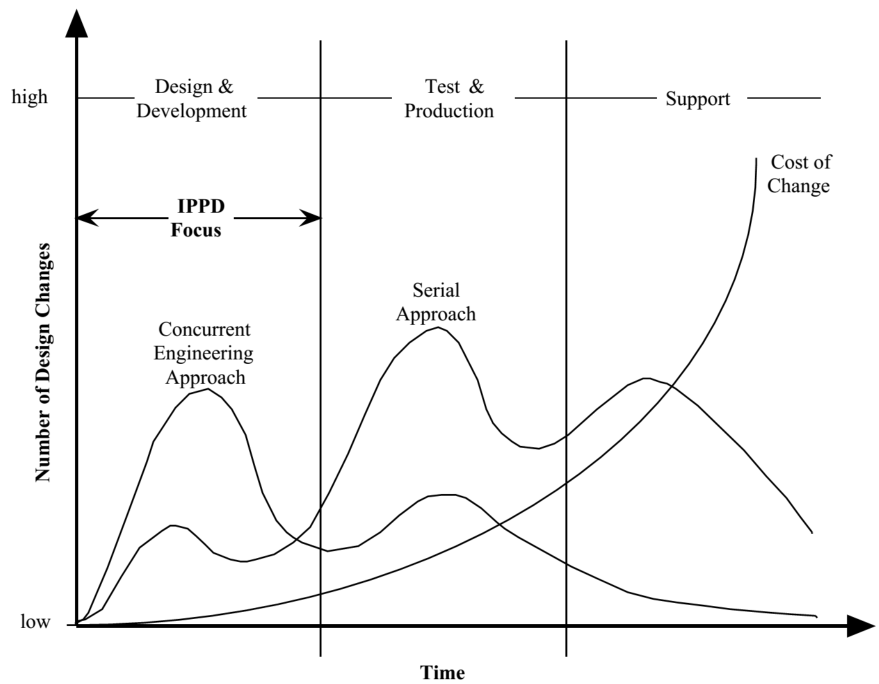


Figure 1: Cost of Change [82]

Committed" curve moving from left to right where most of the final costs are only locked in at very late stages of design. This is accomplished in part by retaining design freedom until later in the design process, shown in the "Freedom" curve which moves from left to right. Both of these goals are dependent on the third set of "Knowledge" curves moving from right to left. The idea is that freedom can be maintained in part by bringing knowledge earlier in the design process to where the information will do the most good.

This thesis is primarily focused on bringing knowledge of the effects of potential actions taken during later design stages forward into conceptual design. This is represented in Figure 2 by moving the knowledge curve to the left as depicted in this image. The premise is that accomplishing this will afford the decision maker more information when making the most important early design decisions.

As a more concrete example, assume an airframe manufacturer, Company Z, has decided to bring a new civil transport aircraft to market. They have performed

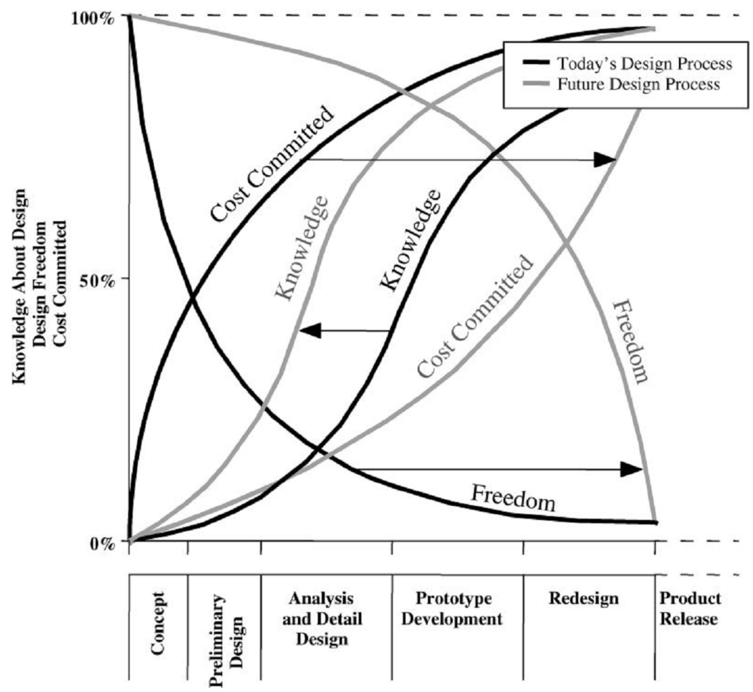


Figure 2: Paradigm Shift [83]

their due diligence in examining the marketplace and are confident of the necessary requirements. Their research has shown that multiple performance requirements exist. Key among these requirements are a design mission, which will require the vehicle to meet a defined long range at a specific payload capacity. Other performance requirements will exist on the aircraft, and Company Z knows it needs to meet these constraints as well.

Initially, little is known about the still theoretical aircraft. As designs are developed, more information decisions are made to further define the details of the new aircraft. To make these decisions, additional information is needed in the form of more detailed analyses. A multitude of decisions must be made in order to completely define an aircraft with enough precision for it to be constructed. Trying to make all of these decisions in a single step would be intractably complicated. Instead, decisions are made first at a very high level and with increasing granularity as design proceeds. These increasingly detailed decisions necessitate additional information, requiring more detailed models.

To implement this, commercial aircraft design is broken up into multiple stages [75]. Typically, more information is known about the design at each progressive stage, and more detailed decisions are made at each progressive stage. This requires more detailed and longer analyses to be performed, as illustrated on the left of Figure 3. As a result of this increasing need for information and increasing time requirements, fewer or smaller scale design changes are generally considered during each progressive stage. Particularly when leaving the conceptual design stage, many of the selected parameters are considered "frozen" - that is to say that they will not be considered for change during later design stages under normal circumstances, reducing the number of dimensions available and reducing the number of designs to consider. This concept is illustrated by the triangle on the right of Figure 3.

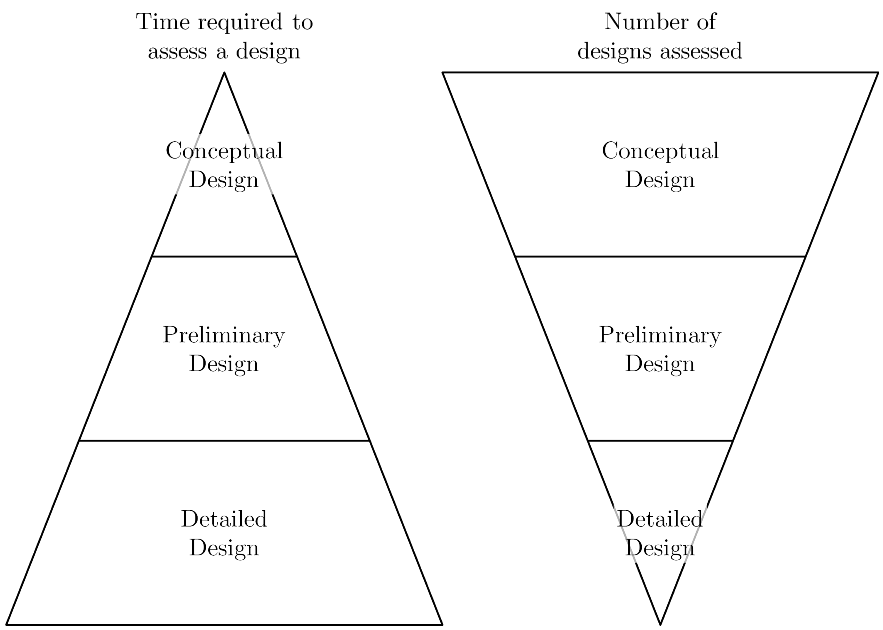


Figure 3: Increasing Time Requirements per Analysis Leads to Fewer Considered Designs in Progressive Stages of Aircraft Development

The process of starting with a low level of fidelity and increasing the fidelity over time has a notable consequence on the certainty of the aircraft's performance. When the gross design characteristics are being selected, the least is known about the aircraft performance. This design uncertainty can lead to the vehicle performance not matching the performance predicted during early design stages. Unfortunately, this is in conflict with the goal of performance guarantees established in early purchasing contracts.

Because a later design stage uses more detailed tools, more people, and has a more concrete design concept, the performance of the aircraft is better known. Rephrasing, conceptual design is less certain than later design stages, meaning that performance estimates will be different from what is realized later. Improperly planning for this uncertainty can lead to a loss in performance relative to expected levels, meaning that a design may no longer meet its performance goals. This performance drop can result in a failure to meet requirements placed on the aircraft, threatening the ability to bring the vehicle to market.

The concept of using uncertainty quantification to enhance aircraft design is not in and of itself original. Zang et al. describes benefits to aerospace vehicles and the barriers encountered when adopting uncertainty-based design methods [96]. They also characterize the then-current state of the art and list avenues in which NASA can advance the field. Zang's work is not unique, as a plethora of efforts too numerous to list exist which have applied uncertainty quantification to aircraft and aircraft component designs [30, 84, 93].

The uncertain aspect of aircraft design has been well-known, and aircraft companies do anticipate these errors. For deterministic design processes, engineers often include uncertainty margins (e.g. a weight penalty over the predicted aircraft empty weight) in their early analyses in an attempt to account for problems which will occur downstream. The application of margins, however, is based on historical information or engineering judgment. Additionally, these margins give no quantification of how reliable the resulting system is - potentially leading to an over-designed product or one with unacceptably low reliability.

Company Z was not ignorant of the inherent uncertainty in aircraft design during previous aircraft designs, but it did not have a method to quantitatively capture the likelihood of meeting the resulting probabilistic constraints. Instead of any quantitative assessment, Company Z imposed conservative margins on the uncertain variables when performing deterministic design to "over size" the aircraft in the hope that this conservatism would account for the uncertain performance. However, Company Z is aware that they had no way to know exactly what level of conservatism would be required.

1.2 Uncertainty Quantification and Management Tools

Engineering risk can be thought of as the probability of an adverse event occurring combined with the consequence or impact of that event. Depending on the relative severity of these metrics, different uncertainty quantification (UQ) techniques can be employed. Formalized UQ techniques fall into two main categories: Robust Design Optimization (RDO) and Reliability-Based Design Optimization (RBDO). In general Robust Design Optimization is concerned with minimizing the sensitivity of performance to outside factors when a degradation is likely but non-critical [87]. On the other hand Reliability-Based Design Optimization is concerned with minimizing the probability of failing to meet a target level of performance because such a failure could be catastrophic [32].

The concept of robustness arises from Robust Design Optimization, a process originally developed by Gen'ichi Taguchi [87]. Taguchi's concept of robustness is primarily concerned with minimizing the loss due to a product varying from its design specification – a situation which will naturally occur during the manufacturing of a large number of products [7]. These losses can be substantial, but are assumed to be generally non-catastrophic to the usability of the product. A robust product is one that is insensitive to variation. Figure 4 demonstrates this concept for an arbitrary objective function.

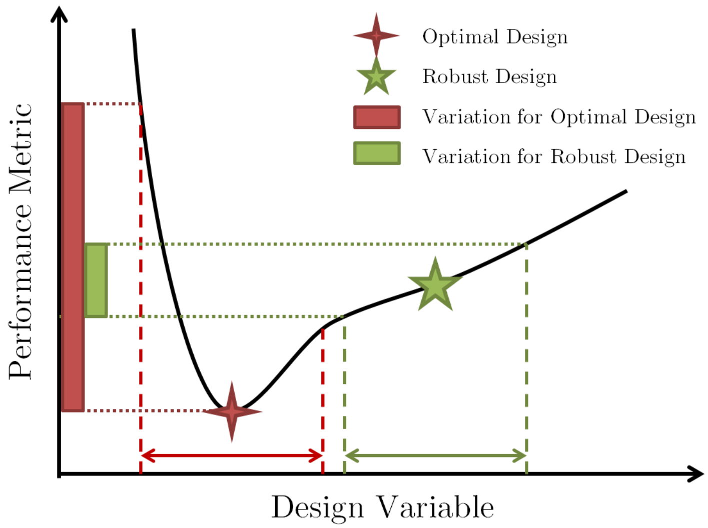


Figure 4: Robust Design Optimization

To enact robust Design Optimization, some form of the optimization algorithm in Equation (1) is followed [95]. A design (x) is solved for which minimizes some function ($F()$), which is a combination of both the mean (μ) and standard deviation (σ). This combination of mean and standard deviation is usually a weighted sum, but can be any form which takes both pieces of data into account.

$$\begin{aligned} \text{Find: } & x \\ \text{Minimize: } & \tilde{f}(x,p) = F\left(\mu_f(x,p), \sigma_f(x,p)\right) \\ \text{ST: } & g(x,p) \leq 0 \\ & x^L \leq x \leq x^U \end{aligned} \tag{1}$$

Reliability is a related but distinct concept from robustness. The core premise of reliability is the likelihood that an item will continue to function, often for a specified period of time [32, 95]. This is often formulated as the ability of the product to meet or exceed its design specifications, regardless of associated uncertainty. This concept is illustrated in Figure 5.

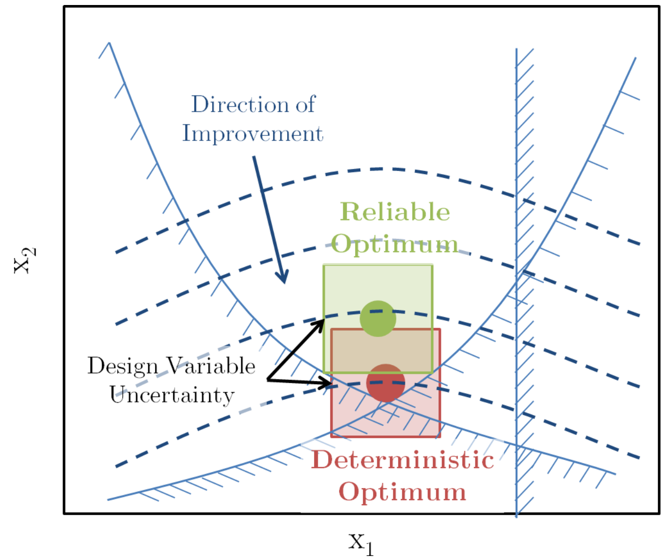


Figure 5: Reliability-Based Design Optimization

Reliability-Based Design Optimization is formulated mathematically in Equation (2). The optimizer searches for a design (x) which minimizes the mean of a metric of interest. The minimization is constrained to keep the likelihood that the design will satisfy any or all constraints ($g()$) above some specified value (R).

$$\begin{aligned} \text{Find: } & x \\ \text{Minimize: } & \tilde{f}(x,p) = \mu_f(x,p) \\ \text{ST: } & P\left(g(x,p) \leq 0\right) \geq R \\ & x^L \leq x \leq x^U \end{aligned} \tag{2}$$

Huyse differentiates the areas of applicability of these techniques in Figure 6 with regards to probability (frequency) and consequence (impact) [40]. Some processes align very obviously with one category or the other. For instance, component performance is often more concerned with Robust Design, since the likelihood of an adverse event is high, while its consequence is usually just a (relatively) minimal loss in vehicle performance [40]. Any safety problem will invariably be suited towards an RBDO-like formulation, based on the very high consequences associated with aircraft failure and the desire to keep the likelihood of those failures to an absolute minimum.

Indeed, Reliability-Based Design largely arose due to structural safety concerns, and a vast number of articles have been published on the subject of RBDO as applied to structural safety [27, 52, 89, 92], as well as spacecraft [84], transonic compressors [46], and car doors [94], among others.

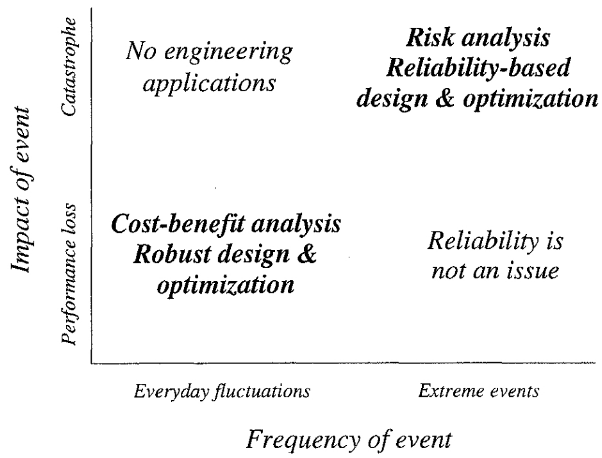


Figure 6: Uncertainty Classification [40]

Methods exist which combine Robust Design Optimization and Reliability-Based Design Optimization into one, such as "R^2BDO" [73]. Methods like these often take the general structure of RBDO optimization in Equation (2) and augment the objective function to more closely mimic the one found in Equation (1).

Vehicle design can fall into either category, RDO or RBDO, depending on the concerns being addressed and the associated assumptions being made. If, in the current environment of carbon emissions concerns and potential regulations, the designer is uncertain as to whether the aircraft will be allowed to fly at its design Mach number or whether it will be forced to fly at a lower one, then designing for robustness to the operating speed of the aircraft would make perfect sense.

On the other hand, if the primary concern of the designer is meeting performance targets such as takeoff or landing field lengths while designing with imperfect prediction tools, then a reliability approach would be more logical. The catastrophe in this case would be to the company's bottom line upon designing an aircraft that cannot fly out of certain airports, rather than a safety concern as with the case of structural design.

Put more generally, the latter set of assumptions looks at cases where redesigning is expensive and missing certain performance targets is unacceptable. As a result, it is desirable for violations of these performance constraints to be infrequent. This thesis is concerned with this second set of assumptions, where performance losses are very unlikely to happen by design, but are of great concern when they do happen.

So under this scenario Company Z is concerned about missing their performance constraints. They may violate established contracts with their customers, potentially resulting in heavy fines or a loss of sales. Such an event could be catastrophic to the financial health of the company. Thus, they consider reliability-based methods to be appropriate for their aircraft design problem. However, they feel that RBDO alone does not account for the complete design process as it has been implemented in the past, as it has can lead to over-sized aircraft with no ability to recover from missed performance targets.

1.3 Missed Performance Targets

Increased reliability cannot be achieved without costs. To increase reliability, the aircraft is designed such that it is more conservative, meaning that its performance is further from the constraints. This conservatism will adversely affect other metrics like gross weight and fuel burn. Increasing this reliability too far will lead to an "over-designed" product, one whose performance has been degraded in excess of what is reasonably required due to excessively high reliability goals.

In other words, there is an inherent tradeoff between the level of reliability and design performance. Because of this tradeoff, it is not always desirable to drive a design to 100% reliability - in fact, doing so could degrade the performance sufficiently that the resulting product may not even be marketable! Thus, even with tools like

RBDO, an aircraft may still fail to meet its performance constraints (i.e.$P_{failure} > 0$). Also, because designing to an extremely high reliability becomes more and more expensive in terms of both actual cost and vehicle performance, increasing reliability becomes a balancing act between spending the money up front when it may not ever be required and spending the money down the line when costs are higher. This balance will be chosen differently for different designers/companies, but will almost never result in a 0% chance of failure.

If a design fails to meet one or more requirements, the aircraft may not be marketable, may fail to meet regulations, or it could be incompatible with airports. Most of these outcomes would result in a loss in sales that could be catastrophic for the company given the amount invested in a new aircraft design. It is undesirable to redesign the aircraft at this later stage because analyses up to this point would need to be discarded, representing a wasted investment. However, it would often be worse to simply give up on the project development or to release an inferior product, both of which could result in a loss of market share.

1.3.1 Mitigation Actions

If a design fails to meet performance constraints, some aspect of the aircraft must change to bring the aircraft into compliance. These changes must affect the missed targets directly, but should also have a minimal impact on prior design decisions. These "mitigation actions" are changes taken to recover a design in the later design stages which has failed to meet one or more requirements. Mitigation actions are separate decisions from the original design choices, and are only taken in the event that they are needed due to the extra cost required. These actions provide an opportunity to leave the original design parameters frozen, while using alternate degrees of freedom to "tweak" the aircraft's performance. However, their application requires an experienced designer to make changes such that new problems are not introduced.

For instance, the reaction to not meeting an approach speed requirement might be to change the wing's high-lift device. However, re-designing this device would make the wing heavier, especially since the wing structure was not originally designed to handle this load. One extreme real world example of this type of balance is presented in Section 1.3.2.

Chief engineers are traditionally responsible for making these finessed late-stage design changes. They possess the necessary experience and authority to take action when the traditional design process has not yielded an aircraft that meets performance requirements. However, the action that can be taken at this stage is necessarily limited so as not to alter the aircraft too much or violate any additional constraints. The particular actions that may be taken vary from company to company.

Raymer describes the possibility of mitigation actions in his traditional aircraft design book. When discussing design "fixes" Raymer mentions that

> "Some of these things, though, are fixes to aerodynamic problems discovered later in design development or flight test. [...] Later on it is too difficult to change the overall geometry, and so if unexpected problems are found, they must be fixed in some other way [75]."

This view of mitigation actions is limited in scope to aerodynamics, but it does describe the concept and motivation of these changes. Specifically, Raymer references the uncertainty inherent in the design process and the temporal nature of reducing that uncertainty. He mentions the discovery of a failure too late in the design process to make gross changes to the design. Motivated by the need to fix this problem without changing the geometric variables, other avenues of mitigation are implemented. Implicit in this statement is the fact that the aircraft resulting from these changes is less ideal than a different design which would have been selected if the problem had been foreseen.

Finding good examples of mitigation actions in the literature is difficult. This is likely due to a number of considerations. If the performance of an aircraft is not grossly outside of desired specification, mitigation actions will be small changes to the vehicle. Additionally, mitigation actions as defined in this thesis are changes that happen primarily during the internal development of an aircraft. These late stage changes may not even be considered a form of mitigation by the design team executing them; they may simply be seen as an expected reaction to common, inevitable problems. Finally, companies are understandably reluctant to publish any details of their proprietary design processes, as the release of such information could potentially aid a competitor by revealing the philosophies of the company.

1.3.2 The 747 and the "Sutter Twist"

While many examples of mitigation are considered proprietary information and difficult to find, there are a few well-publicized examples of late design mitigation actions that had large impacts on the company. Such an example can shed light on the hidden processes in action during preliminary and detailed design. Joe Sutter, the lead designer of the 747, describes one such change in his autobiography, which he attributes to having saved the 747 as a successful aircraft and potentially Boeing as a company [86].

> We had pretty much finished our 747 aerodynamics testing when a crisis erupted over the 747's wing. From an aerodynamic and performance standpoint, we had a wing that worked, but our structures people began getting lots of troubling data from wind-tunnel testing.
>
> Analysis showed that the outboard wing was carrying too much load. The pressure distribution wasn't in the right place for this load to be properly supported by the internal structure. Because of the accelerated pace of development, this realization came quite late in the design process. [...]

> A new wing simply wasnt an option; we would have to fix what was wrong with the current wing. [...]
>
> Hopes rose at the idea of leaving the inboard wing intact and starting the twist only outboard of the outer engine nacelles. If it worked, it would be one tenth as difficult and costly as twisting the entire wing. At my instigation, Jim Hoy performed a rudimentary analysis to evaluate the concept and found that twisting the outer wings would actually yield 80% to 90% of the required benefit of twisting the entire wing. [...]
>
> My people redesigned the 747's wing for an outboard twist, which completely solved the loads problem. This expedient solution worked so well that it got the attention of the press, which labeled it the Sutter twist, a name it still has today. [86]

This example shows a major aircraft development program running into significant "show-stopper" problems very late in the design process. Changing the design at that stage is considered unacceptable, so a different solution must be found to the problem. The solution -twisting the outboard wing- is one which could have been designed into the vehicle initially, but was not considered. These types of changes usually cause the structure of the vehicle to change and increase the weight of the vehicle. Also, there is enormous reluctance to change much of the aircraft because doing so would be "horrendously complex and expensive," but some form of alteration was necessary. Thus, the minimal change required was implemented to meet the constraints, incurring the lowest penalty option available.

So in this use case Company Z wants to consider scenarios like this one when designing their aircraft. While such changes to the design at a late stage are undesirable, they are significantly preferable to scenarios under which the design cannot be recovered. The hope is that by including these "mitigation actions" in an uncertainty process, a better design can be selected which will have the same overall reliability as an oversized design would, with knowledge of how frequently these mitigation actions would be necessary.

1.4 Motivation

As has been described in this section, uncertainty in early design stages can potentially contribute to a design not meeting necessary performance constraints in later design stages. In the real world, should such a problem arise in later design stages, the chief engineer will take necessary actions to bring the design into compliance, if any such actions exist for the failure mode encountered. These mitigation actions are preferred to be of minimal impact to the design choices that have already been made, but are currently not included in early design stage planning. Knowledge of how these mitigation actions could affect a design's overall probability of success through recovery could help distinguish designs with otherwise similar performance and compliance characteristics from one another.

Since mitigation is a response to the presence of uncertainty, probabilistic methods will be necessary to perform this analysis. In order for these design options to be modeled in conceptual design, the following questions must be addressed as part of the design process: How likely is it that Mitigation Actions will be needed? How likely is it that Mitigation Actions will be successful? Can the answers to these questions be additional criteria for choosing designs during conceptual design?

The anticipated outcome of this thesis is the acquisition of a set of knowledge about mitigation actions that can be used to make conceptual design decisions. Among this information is the probability of recovery, i.e. the probability that a non-compliant design can be made compliant using mitigation as well as the expected value of the block fuel, which is a measure of the penalty associated with the mitigation action. With this additional information at hand, certain designs that seemed equivalent before accounting for mitigation may be differentiated by the ability to mitigate them, and if a specific amount of risk is acceptable the design may target this risk level to find the most efficient design at a given probability of success.

1.5 Dissertation Overview

This thesis presents a way to quantify the recoverability of design (the likelihood that mitigation actions will be effective at recovering a design under detrimental uncertainty scenarios) and a methodology for including this information in the conceptual design decision making process. It does so by evaluating the uncertainty space for each design. For points in the uncertainty space which are non-compliant, the mitigation space is investigated in an attempt to find a level of mitigation actions that can make the design compliant. The additional points which become compliant once mitigation is included are the recoverable space, and the probability of recovery is the number of points which have become compliant divided by the number that failed before implementing mitigation. Armed with this information for each design, as well as the expected value of block fuel which was required to achieve the mitigation, the designer would have the ability to trade the cost of mitigating against the additional design performance achieved for that cost.

Chapter 2 contains necessary background information including an overview of aircraft design, a description of uncertainty applied to design, example probabilistic methods which have been used for aircraft, reliability calculation methods, the compatibility between traditional RBDO and mitigation actions, Pareto optimality, and surrogate models. Chapter 3 describes the methodology employed in solving the problem, including relevant sets of research questions and hypotheses, as well as the experiments needed to test the hypotheses. In Chapter 4, canonical problems are constructed to test the posed hypotheses. The results of the hypothesis testing will lead to the formulation of a step-by-step methodology in Chapter 5, developing guidelines for implementation. Chapter 6 develops a specific implementation of the methodology on the conceptual design of a large civil passenger transport to demonstrate the overall methodology. Chapter 7 contains the results of the example implementation, including final hypothesis testing, demonstration of the overall methodology, and trade studies explored utilizing the ideas explored in this thesis. Chapter 8 describes the contributions provided by this work to the state of the art in the field of aerospace engineering.

CHAPTER II

BACKGROUND

This chapter reviews available literature related to the aircraft conceptual design process under uncertainty. By examining the work present, the state of the art of uncertain aircraft design is established. Existing methodologies are examined for their appropriateness to the current problem. Where possible, these implementations will be leveraged. Where no existing work exists within the realm of aircraft design, an attempt will be made to cross-pollinate methods from other fields. If no existing methods are sufficient, the best existing methods will be selected to move forward with and augmented to fit the problem at hand.

First, aircraft design is reviewed, paying particular attention to the stages of design and failure during later design stages due to uncertainty. The concepts of uncertainty applied to aircraft design are examined. This is followed by an investigation of uncertainty quantification methods which have previously been applied to aircraft design under uncertainty. Traditional reliability calculation algorithms from Reliability-Based Design Optimization are examined to determine their strengths and weaknesses. The concepts of Pareto optimality and surrogate models are also discussed, as they are leveraged to aid this work.

2.1 Aircraft Design Overview

The aircraft design process is traditionally performed through sequential analyses which increase in fidelity as design decisions are made and design degrees of freedom are locked down. This is necessary because of the scope of aircraft design, which integrates together the disciplines of aerodynamics, propulsion, controls, and structures among others; the decisions made in these different areas interact and are dependent upon one another making the entire problem impossible to solve at once. By breaking the process into sequential steps through which more is known at each iteration, these interactions can be reduced by assuming values that will be computed later in more sophisticated analyses. Often these assumptions are based on historical data which should theoretically provide a value somewhere near that which will be computed later.

Aircraft companies regulate this process by scheduling design reviews at critical points. These reviews examine the design choices made and provide buy-in from the company's ultimate decision makers. The approved design choices are then fixed as the design moves forward.

2.1.1 Stages of Design

Sources describe the stages of aircraft design differently. Raymer describes a three stage process of conceptual, preliminary, and detailed design, followed by Fabrication as a separate process as illustrated in Figure 7 [75].

Every aircraft company goes through a conceptual design phase. During this phase, many aspects of the vehicle will be in flux and little-to-nothing may be finalized. Wing geometry, weights, cabin layout, the engine, and many other aspects may still be up in the air. However, some aspects are often predetermined. The vehicle's passenger load, payload requirements, design range, takeoff and landing lengths, and many other aspects can be fixed based on federal guidelines or market studies. During design, some of these parameters can be treated as input variables, but many cannot. Thus, the design is more than simply inputting the known parameters and letting equations determine the vehicle's characteristics; instead, the vehicle must be iteratively analyzed using the fuel required to meet the range requirement and fuel available based on the overall size of the aircraft. Once these two quantities match, the aircraft is considered sized to the design mission.

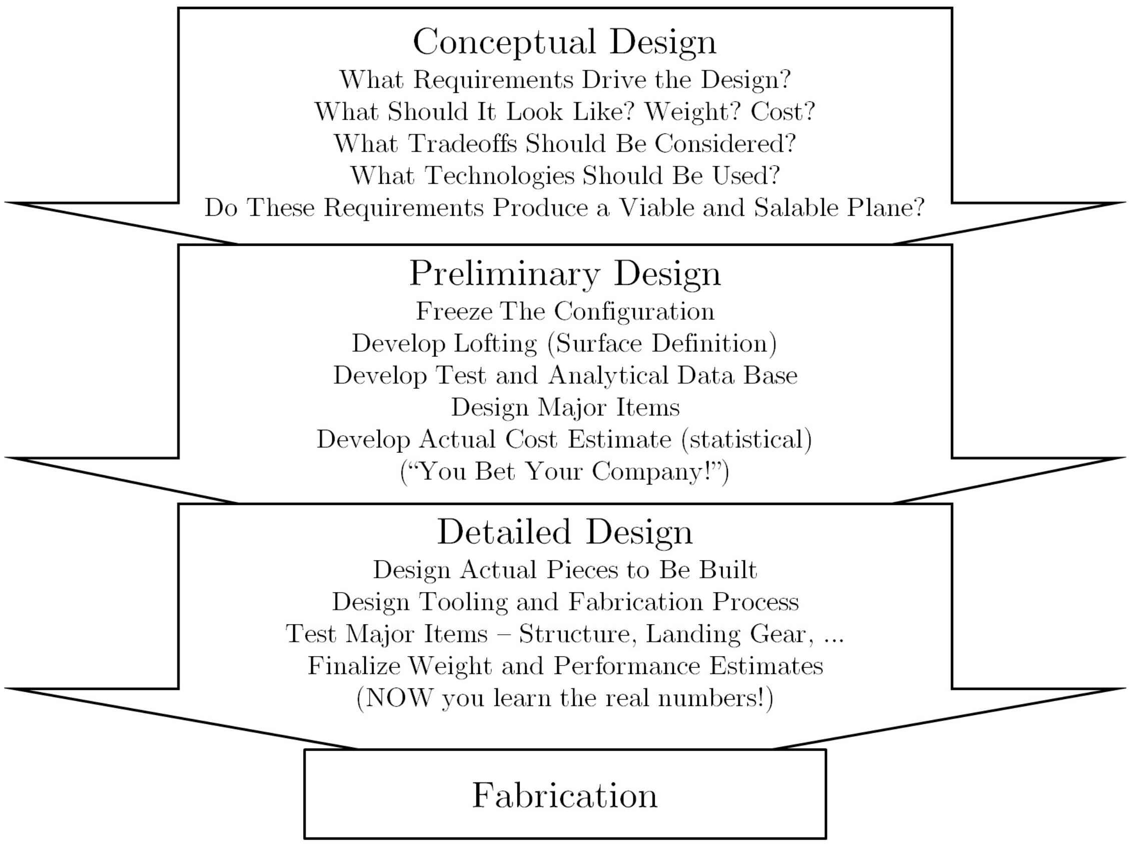


Figure 7: Stages of Design [75]

Figure 8 shows a deterministic design space. In this design space, two design variables are shown, x_1 and x_2. Using aircraft analysis codes, the performance of designs constructed with a particular combination of design variables can be assessed. These performance characteristics can be compared to aircraft requirements or constraints to determine whether the vehicle will be compliant with all constraints. For this conceptual diagram, three constraints are shown via blue curves. The hash-marked side of these lines indicate that further change of a design variable in this direction will cause the constraint to be violated. These constraints restrict the available space for design selection to the region on the non-hashed side of all of the constraint lines. The remaining space is available for design selection. Assuming an objective function were imposed on the design, an optimization could occur to select a design point. This theoretical design point is shown by a blue dot.

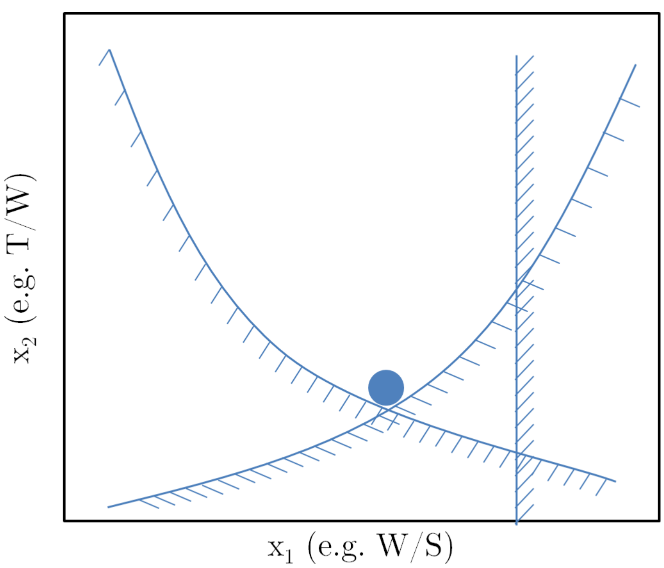


Figure 8: Deterministic Design Space

Eventually a company will settle on a concept with which to move forward. The design up to this point is considered "frozen" and the vast majority of decisions made will no longer be considered for change. At this juncture, the design enters the preliminary design stage. During this phase, engineers use more detailed tools and analyses to refine their assessments of the component designs. Keane and Nair point out that the refined analyses and increased engineering team "severely restricts the number of different configurations that can be considered during preliminary design [44]." Thus, the gross geometry of the vehicle and its overall weights will remain fixed while these more detailed analyses are implemented. During these refinements, the estimates from conceptual design may prove to be incorrect. The vehicle drag may be higher or lower than anticipated; component weight estimates may be incorrect; the engine may perform better or worse than expected.

Aircraft companies do anticipate these errors and have a set of tools to address them: uncertainty margins. Uncertainty margins are commonly used during conceptual design to provide "wiggle room" in the decisions made. These margins are often an attempt to account for uncertainty when the aircraft design process is only executed deterministically [58]. Uncertainty margins can be implemented by making more conservative assumptions about the uncertain design values during the conceptual design process. This conservative assumption will affect the performance of the aircraft

The need for a uncertainty margin can be illustrated by a simple thought experiment examining only a single performance constraint and a single uncertainty variable. The design mission range of an aircraft – a possible performance constraint – is highly dependent upon the aircraft drag, among many other things. If the drag prediction capability of the conceptual design tool is not precise as illustrated in Figure 9, then the performance of the vehicle will vary. Since drag negatively impacts aircraft range, an increase in drag from the design predicted value will degrade the range of the created vehicle. If no capability is implemented to oversize the deterministic vehicle (e.g. a uncertainty margin), the designer will have little-to-no ability to influence the reliability of the aircraft.

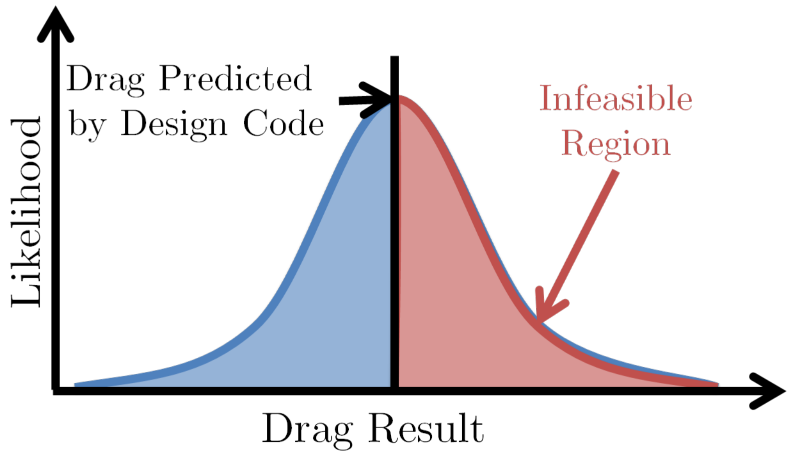


Figure 9: Uncertainty without Uncertainty Margins

An alternative kind of "margin" takes the form of a constant additive or multiplicative factors on certain constraints (as with structures [98]). This kind of "performance target setting" or "probabilistic safety margin [18]" is shown conceptually in Figure 10. This form of target setting has been implemented for aircraft conceptual design under uncertainty [62].

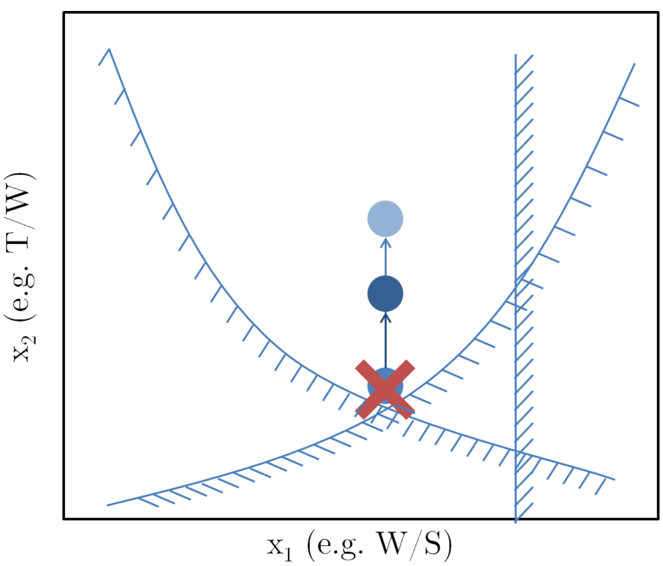


Figure 10: Deterministic Design with Target Setting

Unlike performance target setting, uncertainty margins do not affect the performance of the vehicle by a consistent push-off factor. When the physical implications of this margin are processed through a performance assessment, the margin may cause a different level of impact on the performance depending upon the design condition being investigated.

After conceptual design, uncertainty margins are often used during the aircraft design process as a way to deal with problems which arise. For example, during preliminary design the wing structures team may determine that the previously estimated structural weight of the wing has been exceeded now that more detailed analyses have been performed. In such a case, the Chief Engineer may "allocate" a portion of the empty weight uncertainty margin to the wing structures group as a bookkeeping measure to keep the aircraft's weight and performance on target.

Unfortunately, margins are generally determined by design experts or historical values and are not compared to a quantitative calculation of the reliability which results from the margin application [65]. A problem with this approach is that it is difficult to anticipate how much margin will be needed for a given design. The margins necessary will depend on the selected design point, the performance constraints imposed on the design, and the impact that any uncertainty has on these constraints. No evidence has been found of an aircraft design study accounting for both design uncertainty and uncertainty margins.

Quantitative selection of a uncertainty margin is important because if a margin is not selected appropriately, there will be some cases a selected margin may be more than sufficient, causing the vehicle to be considered "over designed" because a more "sporty" vehicle could have been selected with smaller margins and still meet requirements. The next section talks about the opposite circumstance, in which the uncertainty margin is insufficient to account for the effects of uncertainty.

2.1.2 What Happens When a Design Fails?

Sometimes margins are insufficient to account for unforeseen inaccuracies. Keane and Nair mention that because "the design freezes early on during the design process, [...]

even when the design team knows that the design suffers from some shortcomings, it is often very difficult to go back and change it [44].” It is theoretically possible to return back to conceptual design and start the design process over; however, it is unlikely that a company would prefer to do so unless all other options have been exhausted. The process of taking an aircraft through the design stages requires money, time, and effort. It seems reasonable to assume that a company would not simply throw away its investment unless it was convinced that the design was not salvageable. Understandably, most companies attempt to have very few changes happen to the aircraft later in the design process. As mentioned by Anderson, “If major changes were demanded during this stage [preliminary design], the conceptual design process would have been seriously flawed to begin with [2].”

Instead, it is in these cases that small late-stage changes, mitigation actions, may need to be applied by the chief engineer to recover the design and associated investment without exorbitant additional resources. This kind of situation is exactly what Sutter described in the quote in Section 1.3.2 when talking about problems encountered during the 747’s development. Stanley Kandebo also references similar actions taken by Pratt & Whitney when they were developing the F135 engine for the Joint Strike Fighter [43]. Quotes from the director of the F135 programs, William Gostic, show how the product can continue to be adjusted even after system and development demonstration engines have been tested.

> With the [STOVL] version of the JSF battling weight issues, any additional thrust that can be generated by the engine, as well as any chance to save weight, will be closely scrutinized. “Thrust is a big issue and is especially important for the [STOVL] powerplant” Gostic said.[. . .]
>
> “Right now we have a plan to get the [STOVL] engine to the target weight, and have identified a number of things to actually get the engine below the target,” Gostic said.

It is apparent that these mitigation actions exist in the real design process of an aircraft; however, they are very rarely modeled in the conceptual design process. The goal of this thesis is to model them within this process with a probabilistic approach. Further, it will be shown that knowing the impact of these mitigation actions during conceptual design can help designers select a better, more flexible aircraft.

The presence of uncertainty in the design process requires a shift in how the design process is modeled from the method used for deterministic outcomes. Since a model must emulate a process in the real world, this process dictates how uncertainty should be integrated into the design model. However, different types of design follow different processes. The first one examined experiences uncertainty around the assumptions made when designing the aircraft. The second experiences uncertainty around certain design parameters which cannot be computed in the early stages of design. The two types of studies are similar in their components, but they operate in a different order. To describe them adequately, a clarified description of deterministic aircraft design modeling is needed.

2.1.3 How Deterministic Design is Modeled

Aircraft design can be modeled using two processes that are often used together to evaluate a particular design. These two steps are usually performed in tandem during aircraft design studies. Sizing occurs first to define the final design of the vehicle. Afterwards, this vehicle is input into a performance analysis tool to evaluate any additional off-design conditions.

Eventually, design will progress into more detailed analysis. It is too expensive to leave the design completely open beyond that point because a company would spend vast amounts of resources using detailed methods to analyze designs which would not be used. Thus, a company will freeze the design and then apply high-fidelity tools to analyze it further. During this later phase, the overall aircraft design (especially the gross geometry) will likely no longer change, but more information about the aircraft will become known. Thus, the uncertainty about the performance of the vehicle will decrease, but the design parameters will remain the same.

2.1.3.1 Sizing

Sizing is the act of designing a vehicle to meet a set of performance constraints, usually based on a design mission. Sizing can be accomplished by iteratively adjusting sizing variables (usually the vehicle's weights) and evaluating the resulting aircraft with a performance analysis. This means that a sizing tool can be created by combining a performance analysis with an internal optimizer. Aircraft design tools use this form of iterative application of a performance analysis to size a design to a particular set of conditions [59].

Sizing takes the input of a goal parameter, commonly aircraft design range, and seeks the appropriate fuel weight to accomplish that mission. This is often referred to as a "fuel balance [78]." During this process, a sizing tool will select a guess of the aircraft's overall weight and dimensions. That aircraft will be used to determine its fuel capacity or $fuel_{available}$. The vehicle is then flown through the required mission analysis to determine how much fuel was needed to complete the mission, $fuel_{required}$. With this information, an updated takeoff gross weight (TOGW) will be calculated, and the process repeated until the fuel volume meets the needs of the mission ($fuel_{available} = fuel_{required}$).

A decrease in the weight of fuel required will lead to a reduced empty weight of the vehicle, further reducing the fuel required. This "snowball effect" will allow the vehicle to find the minimum size capable of accomplishing the mission while meeting all other design inputs. The process described above amounts to a performance analysis of the aircraft at each step of the iteration - essentially, each step in a sizing analysis is in effect a different aircraft, where only the final aircraft is selected because the rest violate the laws of physics or are oversized to the design mission.

Changing the assessment condition will result in a different vehicle because the optimizer will react to the new condition and size the vehicle differently. Obviously, this would be the expected behavior if the design variables are altered, as the "design" would be different. Changing the sizing mission range or payload will also have an important effect. It is worth noting that other assumed parameters can have a strong impact on the final vehicle size. Changes to the vehicle's drag polar, component weight factors, cruise conditions, and many others will have an impact on the final size of the vehicle.

The process of sizing is not exclusive to aircraft design. For example, after thermodynamics have been assessed, turbofan engines go through an almost identical process in which components are scaled photographically to meet a thrust target while keeping the thermodynamics constant [91]. For the purpose of this thesis, sizing will be defined as any process during which many design parameters are scaled through a common variable to meet a specific (performance) requirement while maintaining the other design variables in a non-dimensionalized format.

2.1.3.2 Performance Analysis

Performance analysis is a step in which a previously sized aircraft is analyzed for a prescribed mission, usually a mission of secondary importance to the sizing mission. During this performance analysis, the aircraft itself does not change and is considered fixed, meaning the design parameters, component weights, drag polars, and other characteristics do not change. In many cases, performance analysis uses the vehicle's fuel weight as an input and outputs the aircraft's resulting range.

Unlike with a sizing analysis, performance analyses need not always specify a particular mission. Instead, an analysis can be performed to assess the vehicle's performance for individual flight condition. For example, an aircraft's final approach speed to an airport is dependent on only the vehicle's geometry, aerodynamic characteristics, and its landing weight. Thus, an entire mission does not need to be specified to measure this condition.

A key distinction between performance analysis and the sizing iteration discussed in Section 2.1.3.1 is that during performance analysis the vehicle's overall size remains fixed. Neither vehicle geometry nor maximum gross weight will not adjust to meet any constraints imposed upon the aircraft, and the aircraft will remain as the user specified. This distinction will be important when considering the question of how to model the separate stages of design under uncertainty.

2.1.3.3 Deterministic Constraints

For deterministic design, sizing codes may already have constraint analysis available as settings with an internal optimizer [59]. To be consistent with optimization literature, individual constraints ($g_i(x)$) will be constructed with the form seen in Equation (3).

$$g_i(x) \geq 0 := (y_i(x) - y\_req_i) \geq 0 \tag{3}$$

Where $y_i(x)$ is a performance metric evaluated for a specific design, x, an uncertainty scenario, and y_req_i is the performance limit associated with performance metric y_i. This form can be handled by a plethora of available optimizers. Obviously, these constraints will impact the selection of design settings. However, this deterministic formulation only ensures that constraints will be met for a real design if the final vehicle performance is equal to or better than what is predicted during conceptual design or if the margins, which were not set via any reliability analysis, happen to be sufficient.

2.2 Uncertainty in Aircraft Design

A discussion of uncertainty would be incomplete without a discussion of the kinds of uncertainty that may be faced. The risk community breaks uncertainty into two categories: aleatory and epistemic uncertainty [35, 69]. Aleatory uncertainty is related to events which are inherently random; as a result it cannot be reduced, only (sometimes) predicted. Epistemic uncertainty, on the other hand, is related to a lack of knowledge about the process being observed. This description is exactly like the uncertainty that has already been discussed for aircraft design. Since this thesis focuses on values which will become known at a later point in time but must be assumed in the early stages of aircraft design, its purpose is to account for the epistemic uncertainty surrounding these design decisions.

Considering the modeling described in Section 2.1.3, it will be illustrative to consider how different forms of uncertainty would be modeled using these different processes. Explicitly, at what location in the process of sizing and analysis should certain types of uncertainty be modeled to achieve a desired effect? This thought is expounded upon in Research Question 1, located in Section 3.1.

Figure 11 shows conceptually the goals of uncertainty quantification. Namely, the ability to select a design point with confidence based on the possible outcomes of the performance of the vehicle.

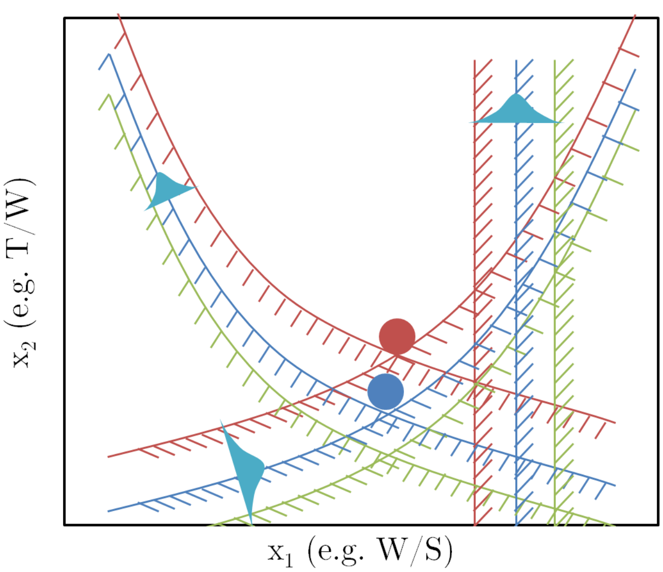


Figure 11: Design Space with Confidence

Daskilewicz et al. point out three possible sources for the kind of uncertainty being modeled here [24]:

- Analysis approaches that employ simplified physics and empiricism in conceptual design;
- higher abstraction and incomplete definition of the intended air vehicle geometry in conceptual design; and
- variability or addition of features in the preliminary design, detailed design, and production phases that were unmodeled in conceptual design.

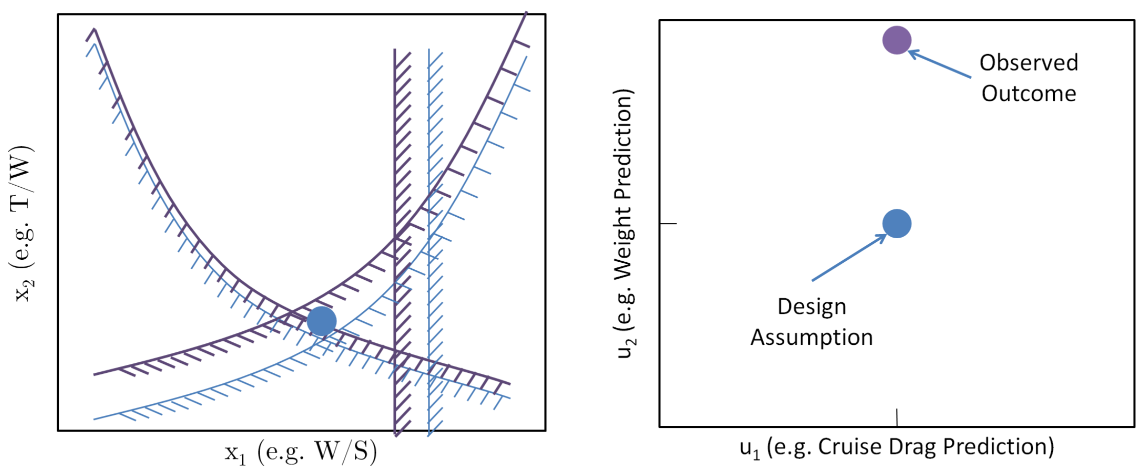
2.2.1 Performance Constraints

Figure 8 illustrates the idea of constraint analysis as it is traditionally used in conceptual design. The goal is to choose the most efficient (usually the lightest) design that meets all requirements. However, Figure 11 is a more realistic depiction of the design process, in which the exact values of design parameters which will meet constraints may not be known. For a deterministic design, the best design is clearly one that meets the constraints while minimizing the objective function. However, design uncertainty may cause the selected vehicle to miss its performance constraints when analyzed in detail in later design stages. Thus, for the design process including uncertainty, the best design point is not as clear as for deterministic design since each point inside the "fuzzy" constraint region will have a different chance of satisfying constraints and a different performance; these two metrics are usually in opposition to one another.

Obviously, if the design is selected well, there will be scenarios –in this formulation, outcomes of the uncertainty variables– in which no constraints are violated. These scenarios are useful information for the designer, but this falls squarely in the well-researched realm of Reliability-Based Design Optimization as described in Section 2.4.

Aspects of this method, will be borrowed and augmented for the current work, but more interesting to the current problem is analyzing what happens when a design fails.

Figure 12 shows the constraints in the design space for a single uncertainty outcome, dubbed Scenario A. On the left (Figure 12(a)) the design space is illustrated with a deterministically designed aircraft at an assumed value of the uncertain parameters. The right side (Figure 12(b)) shows this uncertain parameter assumption. An uncertainty scenario other than the design condition, scenario A, is then considered. This new uncertainty scenario changes the performance of the vehicle. The impact of this performance change is seen in Figure 12(a) as a shift in the constraints within the design space.

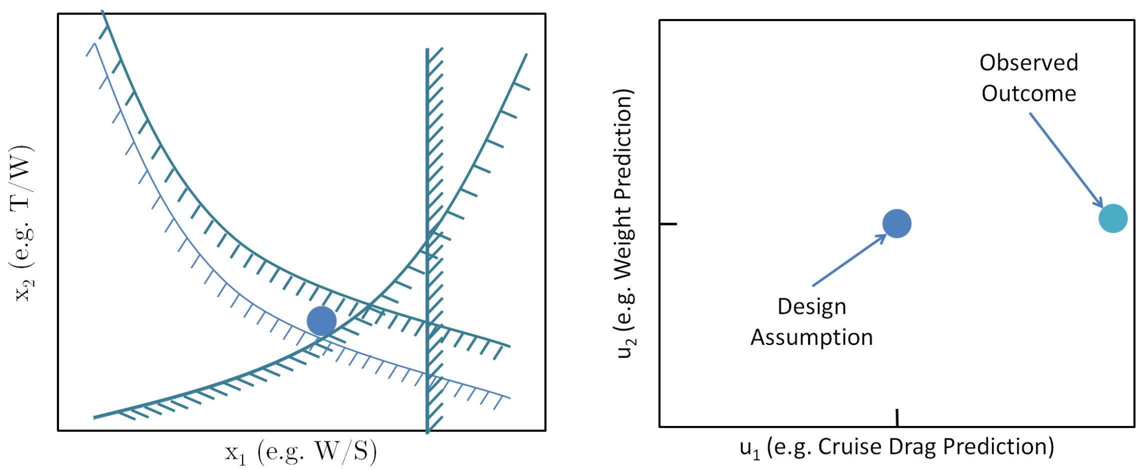


(a) Scenario A Design Space (b) Scenario A Uncertainty Space

Figure 12: Uncertainty Scenario A

The aircraft design was selected at a particular assumption of the uncertainty values and fixed at the end of conceptual design. Later, in the preliminary design stage, it is observed that empty weight of the vehicle is much higher than anticipated. This would have an effect on the performance of the vehicle, moving the constraints in the design space. It is expected that some constraints would be affected more than others. The vehicle may now be violating constraints, depending on how different the outcome was from the assumed design conditions.

Figure 13 shows the constraints in the design space for a different uncertainty outcome, Scenario B. Once again, the left chart (Figure 13(a)) illustrates the design space while Figure 13(b) on the right demonstrates the uncertainty space. Whereas Scenario A resulted in a higher empty weight than was estimated in conceptual design, Scenario B results in a higher cruise drag value than was predicted. In this case a different constraint may be affected than was affected in Scenario A, causing a different constraint violation than in the previous scenario.



(a) Scenario B Design Space

(b) Scenario B Uncertainty Space

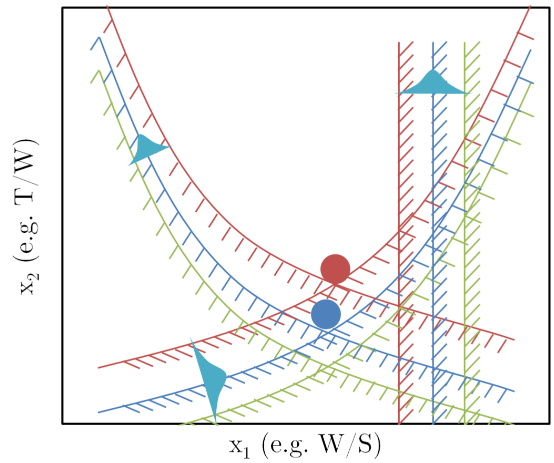
Figure 13: Uncertainty Scenario B

Where Scenario A resulted in a higher empty weight than was estimated in conceptual design, Scenario B results in a higher cruise drag value than was predicted. In this case a different constraint would be affected than was affected in Scenario A. At minimum, it is reasonable to assume that they would violate the constraints by different amounts. If the design must be brought to the same minimum level of performance with respect to the constraints, then different mitigation actions, described further in Section 2.2.2 would be taken in the preliminary design stage to remedy these two different scenarios, even though both started with the same design point.

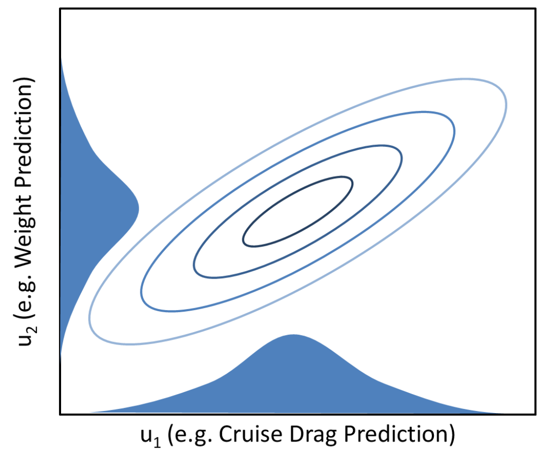
Uncertainty quantification tools attempt to look at all possible scenarios. This can be done via brute force methods like Monte Carlo Simulation or a variety of more sophisticated tools like those described in Section 2.4. However it is accomplished, these methods and others like them aid the decision maker in selecting an appropriate push off from the deterministic constraints to allow for an acceptable level of reliability. This concept is illustrated in Figure 14. Figure 14(a) shows a representative aircraft design space with the deterministic design requirements illustrated by the blue hashed lines. The resulting design point is shown by the blue dot. Figure 14(b) on the right shows the space of uncertainty variables. A correlated distribution has been imposed on these variables, indicated by the distributions on the axes and the concentric, angled contour lines in the chart itself. For each different scenario in the uncertainty space, the constraints in the design space will move. By assessing this entire distribution of uncertainty variables, probabilistic locations of the constraints in the design space can be located. In the design space, this is conceptually shown by the distributions on each of the design constraints. The green constraint lines correspond to a best-case scenario of where the constraint lines could end up, while the red constraint lines represent a worst-case or a bad but unlikely scenario. If one were to design with reliability as the primary goal, a conservative design point like the one shown by the red dot might be selected.

2.2.2 Mitigation Actions

When the design fails to meet one or more constraints due to changes in the predicted capabilities of the aircraft, the chief engineer will step in. As mentioned in Section 1.3.1, mitigation actions are designed to respond to these performance shortfalls. Figure 15 shows how this could work conceptually for a scenario which violates a constraint. Figure 15(a) on the left show the design space with a constraint violation caused by some uncertainty scenario, just like the one observed in Figures 12 and 13. Figure 15(b) show the available mitigation space when the constraint is



(a) Design Space with Confidence



(b) Uncertainty Space for Confidence

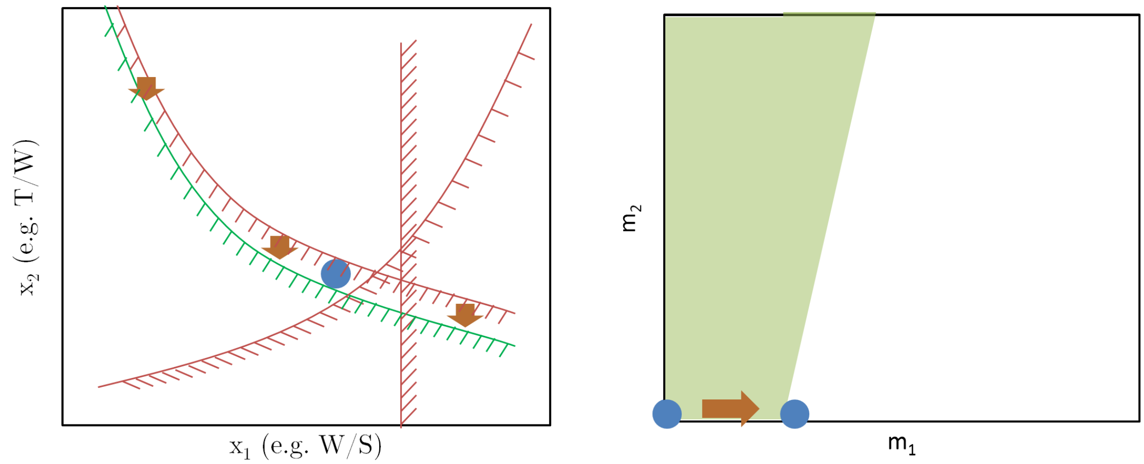
Figure 14: Design with Confidence Bounds

violated. Just like with the design space, the mitigation space is impacted by the constraint violation, reducing the available space. The origin in the mitigation space represents the behavior of the design under the selected uncertainty condition when no mitigation action has been imposed. The fact that this portion of the mitigation space violates the constraint simply implies that the design will not be compliant with the constraints when no mitigation action is used. The open white space in the mitigation contour plots show the available mitigation actions which would bring the design back into compliance.

Moving to this available level of mitigation is equivalent to applying a mitigation action during preliminary design will bring the design back into compliance with the violated constraints. The reaction to this applied mitigation action will be seen in the design space as the constraints moving with respect to the design point. Since the scenario is now compliant with the constraints, this scenario is considered “recoverable” through the available mitigation actions.

It is possible that the mitigation action will not have enough effect to alleviate the constraint violation. In this case, the scenario will be unrecoverable.

If the mitigation action were only to affect the intended response with no other



(a) Deterministic Design with Constraint Violation

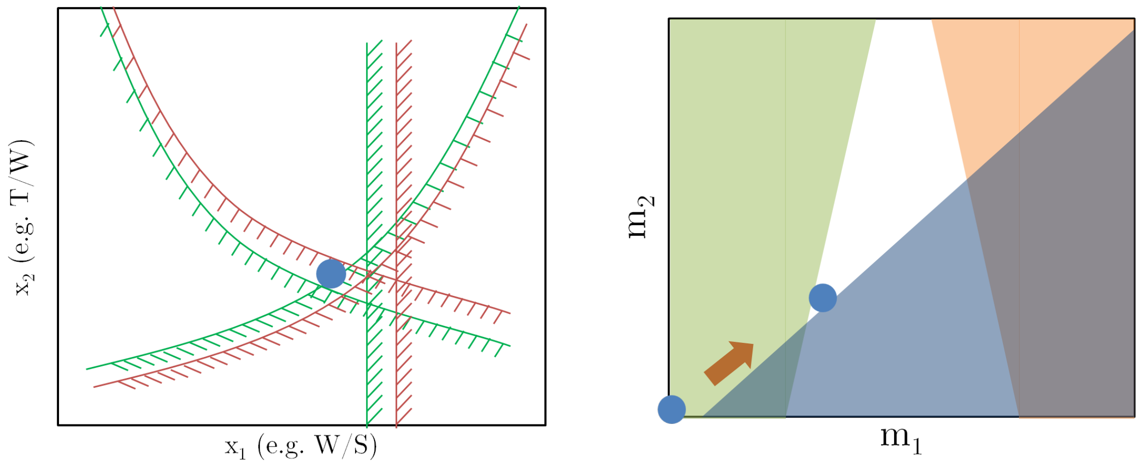
(b) Mitigation Space for Constraint Violation

Figure 15: Constraint Violation

consequences, then this would be wonderful. However, the example from Sutter cited in Section 1.3.2 implies that this is not the case. Indeed, it is assumed that most actions will incur some form of engineering penalty. If there were no penalty or negative consequences to mitigation actions whatsoever, then the action would likely have already been implemented before preliminary design, regardless of any uncertainty considerations. At minimum, mitigation actions will likely increase the empty weight of the vehicle, either directly or through the need for a reinforced structure due to the changing design.

These penalties will have a direct impact on the mitigation space. Consider a constraint which is highly dependent on the empty weight of the vehicle. If the mitigation action employed (m_1) in Figure 15(b) were to have a large empty weight penalty associated with it, application of this mitigation action may cause another constraint to be violated. Figure 16 shows how the mitigation space could change with mitigation penalties imposed.

It is easy to imagine a scenario where a design is already very close to a (non-violated) constraint for a particular scenario. In such a case, mitigating may cause that design to violate the other constraint before enough mitigation is applied to fix



(a) Deterministic Space (b) Mitigation Space

Figure 16: Constraint Violation with Mitigation Penalties

the constraint originally violated. In such cases, it is very possible that no viable level of mitigation action will exist which could fix the design, even though the original violated constraint may be alleviated. In this case, the scenario is also unrecoverable.

The cost of potential mitigation actions would be helpful in determining the usability of mitigation actions. Unsurprisingly, no information is available in the literature to quantify the cost of potential mitigation actions. This information may be available to an aircraft manufacturer behind closed doors. Instead, the engineering penalty discussed above will be used in place of a financial cost. Practical applications of the methodology proposed in this thesis would do well to incorporate this cost information.

2.2.3 How to Address Uncertainty: Margin vs. Mitigation Actions

It will be helpful to clearly delineate margins and mitigation actions, as while they are very different actions taken by the design team, they are intended to serve similar purposes.

Margin variables (h) can indeed be thought of similarly to design variables - they are set when the aircraft is designed/sized. Mitigation variables (m) are reactions to performance shortfalls, and are used only when necessary. Both of these variable types are intended to address uncertainty, but in very different ways. The financial analogy would be that margins are like saving money ahead of time just in case, while mitigation actions would be equivalent to taking out a loan when things go awry.

Assume that the designer does not trust his code's empty weight prediction because it's a conceptual-level tool and/or the vehicle is very new. He may believe that resulting aircraft might weigh more (which is bad) than the code or less (good) than the code's prediction. To compensate for the possibility that the vehicle may be heavy, he can add an empty weight margin, increasing the weight predicted by the code. Because the aircraft is still being sized, the design will compensate (e.g. the aircraft will get bigger to carry enough fuel to meet the design range). If the designer has set the margin well, he can be reasonably sure that my aircraft will meet its performance requirements. Hopefully, through an uncertainty quantification method, he can then quantify the likelihood. For the sake of this discussion, assume that the designer set the margin and design well, and the resulting design has a probability of compliance of 90 percent.

Now what if, despite the uncertainty margin being in place, an uncertain scenario is encountered which puts the design performance in that ten percent region where the design fails a performance requirement? This scenarios is where mitigation actions will come into effect. A design team would try every trick at their disposal to fix the vehicle and bring it into compliance with the constraint. Mitigation actions are a set of these fixes that are considered acceptable, to some extent or another.

A nuance of this whole setup is that the idea of knowing "how much mitigation action to apply?" for a given design is very complicated, and may not actually be sensical. Mitigation actions are only relevant to the design AND the uncertainty scenario in question, and statistics of the results might lend themselves to very easily incorrect interpretations. Mitigation levels may make some sense if evaluated using conditional probability with respect to the uncertainty scenario being investigated.

This concept is beyond the scope of this study. However, it may be a beneficial area to explore for future work.

2.3 *Probabilistic Methods Applied to Aircraft Design*

This section examines a subset of probabilistic methods which have been previously applied to aircraft conceptual design. These are examined both to assess how prior analyses have been performed and to determine whether any methods can be directly appropriated for use in this thesis.

2.3.1 Robust Design Simulation

Mavris et al. introduced the idea of applying reliability analysis to aircraft systems design with the development of Robust Design Simulation (RDS) [56]. The core of RDS is described in Equation (4). Since all random variables Y are purely economic variables and have no impact on the performance of the vehicle, the performance of the aircraft can be modeled deterministically. The output of RDS is a design (X) which maximizes the probability of meeting a single economic design objective while deterministically meeting all performance constraints.

$$\begin{aligned} &\text{Maximize: } P[Z(X,Y) \leq z_0] \\ &\text{Subject to: deterministic performance constraints} \end{aligned} \tag{4}$$

where Z is the overall economic objective, X is a vector of deterministic design variables, Y is a vector of random economic variables, and z_0 is the target of the objective function.

In order to accomplish this, surrogate models of the responses are generated using a design code. These surrogate models are then used via Monte Carlo Simulation (MCS) to generate a second set of surrogate models, this time of the Cumulative Distribution Function (CDF) of reliability as a function of the design variables. The optimizer in Equation (4) then uses these CDF surrogates in conjunction with the response surrogates to find the design with the highest probability of success.

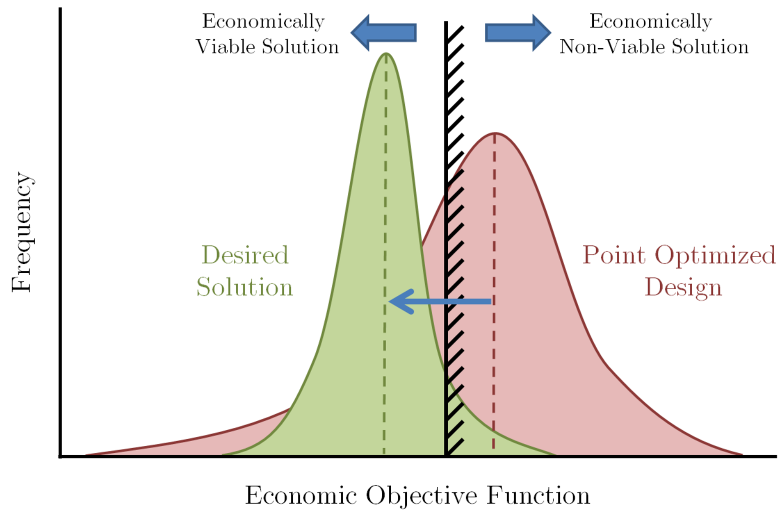


Figure 17: Robust Design Simulation

Mavris and DeLaurentis extended this concept to combined economic and technology uncertainties [57]. However, in this extension, the technology assessment and Robust Design Simulation are performed in separate, sequential steps. Additionally, the assessment method for technology studies is formulated differently than reliability studies, as will be explained in Sections 3.1.1 and 3.1.2.

While its implementation is cumbersome, specifically the surrogate models of surrogate models, the inclusion of reliability as a response to be optimized rather than just as a constraint is novel. This optimization has the potential to counter a pitfall of RBDO noted by Bordley and Pollock in 2009, namely the rejection of a design with much higher probability of satisfying constraints in favor of one with a slightly better value in the traditional objective function [11]. This idea of maximizing the probability of success will be explored again with another tool, Pareto Optimality, in Section 2.5.

2.3.2 JPDM

Obviously, the limitation of RDS of only allowing for a single probabilistic constraint is a significant restriction. Thus, one of the developers of Robust Design Simulation, Bandte, improves upon the work started in RDS by introducing Joint Probability Decision Making (JPDM) [4]. JPDM introduces the probability of success of meeting constraints on multiple criteria simultaneously. This method is used for analyzing aircraft at a system level through one of five different analysis schemes, based on either MCS or Fast Probability Integration (FPI). This reliability measure was used to optimize the design of aircraft.

Both RDS and JPDM use the probability of meeting the specified goal(s) as their only metric for optimization. Additionally, both methods were conceived with the idea of performance constraints being met deterministically, and changing the design to meet probabilistic economic constraints. Their implementations still serve as useful examples of RBDO methods applied to aircraft design.

2.3.3 Stochastic Programming with Recourse

Stochastic Programming is another method of uncertainty quantification which is distinct from RBDO and RDO, but with many conceptual similarities [8, 53, 79]. Particularly of interest to the current problem, Dantzig and Beale first developed a formulation for treating stochastic programming with fixed recourse - a concept extracted from Stochastic Programming - which is considered as a way to fix problems arising from the uncertainty present in conceptual design during later design stages. [23, 6]. Choi examined the concept of Stochastic Programming with Recourse as a way to handle aircraft design uncertainty [20].

Notable differences exist between reliability-based implementations and Stochastic Programming with Recourse. One of the founding principles of Stochastic Programming with Recourse methodology is the assumption that there will always be an available recourse to any unexpected scenario. While any engineer worth his salt will endeavor to find an available solution should a problem occur, it seems unreasonable to assume this will always be the case. The changes to the design may simply be too drastic. Additionally, recourse actions are found in a space identical to the original design variables. While the method proposed in this work does not prohibit such changes to the design, the operating assumption is that the design will not be altered except in those limited ways explicitly considered by the philosophies of the individual(s) executing the method.

Stochastic Programming with Recourse assumes that, aside from the imposed penalty function, the "best" design can be reached after the uncertainty has been resolved. In other words, the designer will re-examine the original conceptual space to determine the best design now that the resulting values of the uncertainty variables are known. The existing design will then be altered to match as closely as possible to this new design selection. The penalty function is imposed to illustrate that, unless the design process is truly reinitialized, this resulting design will retain information from the failing design and thus will be an imperfect representation of the best design for the encountered scenario. This assumption may be acceptable for small changes to the design parameters, but it runs counter to the founding assumptions of this thesis – specifically that the design parameters will generally not be altered after the conceptual design stage is completed. Additionally, the penalty from recourse is not processed through the assessment tool, thus ignoring any secondary effects of the imposed penalty on the aircraft's performance. Thus, Stochastic Programming with Recourse will be used only as motivation in the current work.

2.3.4 Multi-Stage Reliability-Based Design Optimization

Nam et al. applied Multi-Stage Reliability Based Design Optimization (MSRBDO) to aircraft design as a way to deal with the staged nature of the design process [63, 64]. The concept of MSRBDO is based off of traditional Reliability Based Design Optimization with some significant augmentations. Like all reliability methods, MSRBDO attempt to address the issue of making design decisions under the presence of uncertainty. Traditional Reliability-Based Design Optimization assumes that all decisions are made very early in the design process, long before the uncertainty is reduced. Multi-Stage RBDO alters this assumption based on the principle that a longer developmental time will allow the designer to delay some decisions until a later point in time. At this later point, some but not all of the uncertainty will be reduced through additional analyses. At this point more design decisions must be finalized, with a reduced amount of uncertainty. This concept is illustrated in Figure 18.

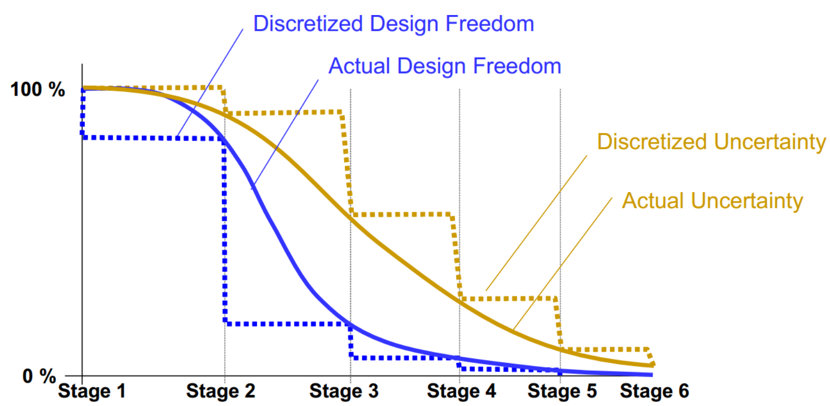


Figure 18: MSRBDO Simulation of a Complex System Design Problem [63]

MSRBDO assumes that this ability to delay design decisions is a natural process of the staged nature of long development time designs like new aircraft. Indeed, many decisions about the specifics of an aircraft are left until later design stages, including after the design freeze. Some such decisions remain because the limitations of conceptual stage design tools do now allow for the finalization of all detailed decisions.

MSRBDO assumes that no penalties are incurred by making these decisions at a later stage. These decisions were delayed because they did not need to be finalized at the conceptual design stage. Instead, additional analyses are performed. These analyses will likely be assessing more fine changes to the aircraft configuration. In this case, more detailed analysis tools will be required. These tools require more time, energy, and more information than conceptual design tools. MSRBDO requires these advanced stage analysis capabilities. If one were to implement full MSRBDO, analyses would need to be performed at all levels for each instance of a design decision, leading to an explosion of assessments while using highly sophisticated tools. In the time required to perform such an assessment, multiple aircraft could be designed in detail.

MSRBDO is founded on three key assumptions [63]:

1. Reducible Uncertainty: Designers have accumulated some knowledge, which also means cumulative reduction in uncertainty has been achieved from the previous work.

2. Retained Design Space: Designers hold a certain degree of design freedom that can be exploited to correct the decisions made at previous stages with increased knowledge.

3. Decision for Feasibility: Designers determine the present stage variables such that the committed design combined with anticipated decisions at ensuing stages guarantees a target feasibility of the present stage.

The example problem discussed in Chapter 1 maintains the first assumption: uncertainty will be reduced at some later point in time during the design process. However, the second assumption does not hold for this problem. In the motivating problem, it is assumed that most or all of the uncertainty will not be reduced until after the design has proceeded into later stages when more detailed analyses will be employed. Since the design is assumed to be "frozen" at the end of the conceptual design stage, none of the conceptual level design parameters should be changed at this later stage. The only time any changes should be implemented late on would be when performance has fallen outside acceptable constraints. The design should only be remedied by a small subset of mitigation actions to recover the performance. Additionally, the third assumption is somewhat true and somewhat false. the designer will still select the best design at the current stage that will still meet requirements later down the line; however, the later design decisions are not intended to occur unless absolutely necessary.

The aircraft example problem used by Nam to demonstrate two-stage MSRBDO (TSRBDO) on an aircraft design process illustrates some of the practical limitations of this method [64]. The design problem does not incorporate any sophisticated pre-existing sizing analysis into the method. Further, though MSRBDO is motivated by the concept of later design decisions being made after more detailed analyses have reduced the uncertainty surrounding the design, no such detailed analyses are included. Nam cites his reasoning for

Nam states that one of the potential flaws of MSRBDO is the need for detailed analyses. These detailed analyses are generally considered inappropriate for conceptual design because they take significant time to analyze a proposed design during a time in which many tradeoffs are occurring. the "design and analysis tools that are commonly used during the conceptual design phase do not support high resolution of designs required to capture the impacts of hypothetical design changes that would take place in the detail design phase [63]."

"In addition, computational practicability must also be considered in designing an MSRBDO problem. The computational effort required to solve MSRBDO increases exponentially with the number of stages, therefore becomes prohibitive very quickly [63]."

MSRBDO assumes access to detailed analysis tools and the ability to run those tools in a very quick manner. Further, MSRBDO assumes that there will be enough knowledge known about the theoretical design to utilize those tools. The analysis tools used during later design stages not only take more time and manpower to execute, they also require more detailed information about the design itself. This level of information is beyond the level of information that typically exists during the conceptual design stage of aircraft development.

For example, Company Z has a wing structural analysis tool that it typically uses during preliminary stage of design. The tool can optimize the wing's internal configuration for minimum weight based on an input aerodynamic load distribution. Typically, this information is given to the structural engineers by the aerodynamics engineering group based on their own assessment of the configuration of the wing. Both of these assessments take significant time. Further, neither of these assessments can even begin to take place until the aircraft's overall configuration is known.

Assessing either of these disciplinary analyses will take significant effort both in terms of time required to perform the analysis and time required to even setup the analysis in the first place. Company Z simply does not perform this level of more detailed analyses during conceptual design. The company will be especially reluctant to perform these analyses for hundreds or thousands of designs.

2.4 Reliability Calculation Methods

Determining the reliability of a design is an important part of assessing the uncertainty of aircraft conceptual design. In order to accomplish this, some method must be employed to assess under what scenarios the aircraft will successfully comply with requirements and under what scenarios it will fail. The field of Reliability-Based Design Optimization is filled with many methods to accomplish this task. Some of the available methods are reviewed below in an effort to determine which is appropriate for the problem at hand.

Reliability methods are often established to measure the likelihood that a design will comply with an individual reliability constraint. It is also possible to determine the reliability of achieving compliance with multiple separate constraints simultaneously (aka joint probability), depending on the reliability calculation method employed. Further, it is even possible to formulate a methodology which employs different reliability goals on individual constraints as discussed in [60].

2.4.1 Sampling Methods

Sampling methods include Monte Carlo simulation, importance sampling, and adaptive sampling. Monte Carlo Simulation (MCS) is a sampling method frequently used for reliability analysis [48, 54, 72]. It is also commonly used in Aerospace Engineering [12, 29, 47].

MCS first draws a set of n random samples ($U = (u_1, u_2, \ldots, u_n)$) from predefined distributions [36]. For each of these samples, a deterministic simulation is employed to analyze the function's response to these input distributions ($y_1, y_2, \ldots, y_n$). For MCS to approximate a function such as the expected value of the input distribution, a simple summation and normalization of the points is executed, as seen in Equation (5)

$$E(y) \approx \frac{1}{n} \sum i = 1 n y_i \tag{5}$$

A key advantage of Monte Carlo Simulation is that it that the accuracy of the results can be specified by the number of simulation runs used [95]. Regardless of the number of dimensions being assessed, it can be shown that the error of probabilistic estimations made using Monte Carlo simulation will reduce approximately with the square root of the number of assessment points. Thus, it is commonly used as a point of comparison for new methods within the reliability community. Even for tested methods, MCS is often used to check the final result of many RBDO studies.

Sampling methods, and specifically Monte Carlo simulation have a long history of association with reliability analyses either through direct application or error checking [39, 32, 31, 71].

Another advantage of Monte Carlo simulation is the ease with which it can simulate different distribution types. By default, computer random number generators create uniformly distributed variables; however, through transformations, almost any distribution type can be simulated. Additionally, Monte Carlo simulation can also generate correlated random distributions through a Cholesky decomposition [34].

2.4.2 Constraint Approximation Methods

For probabilistic methods which do not employ sampling methods, the integral of the constraints over the uncertainty space must be calculated directly or approximated in another manner [25]. Since the Limit State Function (LSF) – the boundary with active constraints – can be arbitrarily complex, an approximation is usually preferred. The following reliability analysis methods all employ some form of approximation. Because the distribution of the performance is not known, it is often easier to simply the integrand of the uncertainty space. This is accomplished by transforming the original random variables by a Rosenblatt transformation [77]. In the literature, these spaces are called the X-space and the U-space, respectively.

In actuality, a transformation is needed to translate X-space to a U-space for easy integration and then an inverse transformation used on the U-space to acquire the X-space for function evaluations (f and g). In practice, the transformation from X to U is largely irrelevant. For methods like the First-Order Reliability Method (FORM), the U-space will always look the same aside from the number of variables: standard normal distributions. Thus, only transformations for U to X are required, and this can be done outside of the main FORM method (between method and function evaluation). This makes integration easy because the iso-contours are all circular. By design, the probability density function (PDF) of this space is known to be

$$\Phi_U(u) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u_i^2\right) \tag{6}$$

The probability of being in an area is now straight-forward to calculate once the integration boundary is known.

$$p_f = \int \cdots \int_{g(u_1,u_2,\ldots,u_n)<0} \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u_i^2\right) du_1 du_2 \ldots du_n \tag{7}$$

It is desirable for the integration boundary approximation to be the most accurate at the point where it will have the highest contribution to the integral. Because all uncertainty variables now have uncorrelated standard normal probability density functions, the point where $g(u) = 0$ comes closest to the origin will be the the location of the largest contribution to the aforementioned integral. This point is dubbed the Most Probable Point (MPP) of failure [19]. Since the MPP is the point which will have the largest contribution to the probability of failure, it is where the most accuracy is demanded, and boundary approximations will be centered about this point. Figure 19 demonstrates the concept of the MPP pictorially.

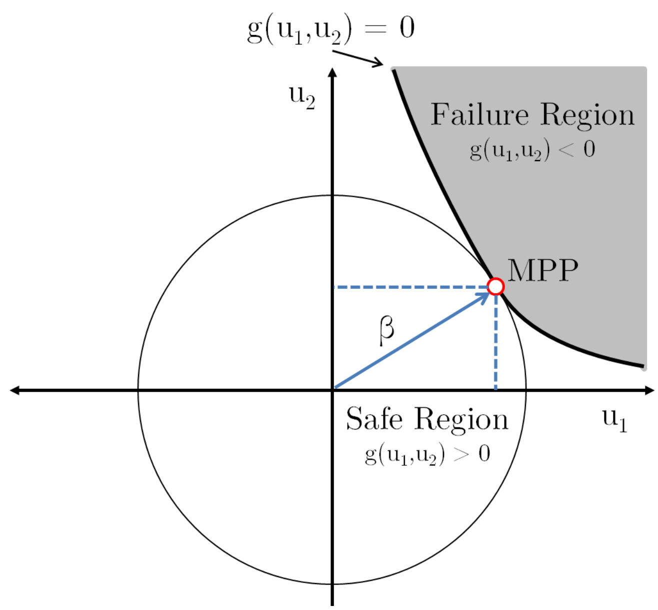


Figure 19: Most Probable Point (MPP) of Failure

Methods to search for the Most Probable Point (MPP) of failure are described in Section 2.4.2. Finding this most probable point constitutes a second 'loop' for any optimization process [3]. This occurs because the act of finding the Most Probable

Point requires its own optimization, separate from the optimization of the design itself. Once the MPP has been obtained, the reliability index β is calculated. This reliability index is equal to the magnitude of the distance between the MPP and the origin in the U-space or $\beta = ||u||$. How the reliability index and the MPP are used differ depending on the method used to calculate the reliability of the system.

The First Order Reliability Method (FORM) is an algorithm approximate the bounds of a Limit State Function (LSF) (i.e. constraint) by first locating the closest approach of the constraint to the center of the uncertainty space [97]. Since this point will contribute the most to an integral over the failure region, the boundary of the constraint is approximated linearly around this MPP. Because FORM assumes the constraint to be linear, it will by definition be tangent to a sphere at distance β, where β is the magnitude of the vector $|u|$, the vector which defines the location of the MPP. Since the entirety of the uncertainty space is assumed to be uncorrelated standard normal distributions, the reliability can be measured using the CDF of a standard normal distribution as in Equation (8).

$$R = \Phi(-\beta) = 1 - \Phi(\beta) \tag{8}$$

Where $\Phi()$ is the CDF of a standard normal distribution.

Obviously, the linearity assumption will be less accurate the more curved the actual constraint is. Also, the method is only designed to handle one constraint at a time. Despite these limitations, FORM is widely used in RBDO literature due to its ease of implementation and relatively few number of function calls compared to higher order methods.

Hasofer and Lind proposed this kind of formulation in 1974 in order to ensure invariance of the calculated reliability with respect to the Limit State Function [33]. Rackwitz and Fiessler furthered this method in 1978 to be applicable to non-normal distributions by incorporating ‘equivalent-normal’ distributions, approximated around the MPP [74]. Hohenbichler and Rackwitz expanded this algorithm in

1981 to be usable for non-normal, correlated random variables in the Advanced First Order Reliability Method (AFORM) by employing the Rosenblatt transformations [76]. The combination of these methods is shown in Equation (9) and is commonly referred to as the Reliability Index Analysis (RIA) [51], the Hasofer-Lind Rackwitz-Fiessler (HL-RF) algorithm [32], or the first-order reliability analysis [90].

$$\begin{aligned} &\text{Minimize: } ||u|| \\ &\text{Subject to: } G(u) = 0 \end{aligned} \tag{9}$$

RIA with HL-RF can be difficult to solve if the LSF (i.e. $G(u)$) is complex and an implicit function, and it does not converge for some problems [49]. Tu et al. introduces an alternate way to locate the Most Probable Point, dubbed the Performance Measure Approach (PMA) or the first-order inverse reliability analysis [90]. This method is formulated as in Equation (10).

$$\begin{aligned} &\text{Minimize: } G(u) \\ &\text{Subject to: } ||u|| = \beta_a \end{aligned} \tag{10}$$

The Performance Measure Approach has many potential benefits over the Reliability Index Approach, especially in terms of the stability of the optimizer due to the spherical constraint of the search area, as opposed to the arbitrarily-shaped constraint of $G(u) = 0$ in RIA. It has the ability to be much more efficient at finding the MPP than RIA, but this is dependent on the constraint in question, thus hybrid approaches are sometimes used instead.

The Second Order Reliability Method (SORM) is very similar to FORM [32, 97]. The method still finds the Most Probable Point of failure and expands the reliability analysis about that point. However, once the MPP has been found, SORM uses a second order Taylor-series expansion around the MPP to find the boundary of integration for the reliability calculation which can be seen in Figure 20.

The second-order approximation more accurately approximates the boundary of integration; however, this does not come without a cost. Because it needs second order

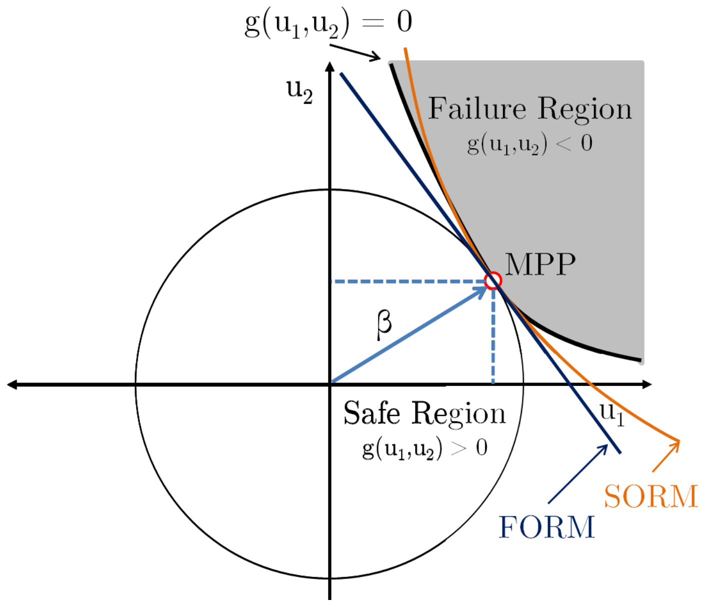


Figure 20: First (FORM) and Second Order Reliability Methods(SORM)

information, SORM calculates a Hessian matrix around the MPP. This calculation is obviously far more involved than the simple $R = \Phi(-\beta) = 1 - \Phi(\beta)$ employed by the First Order Reliability Method, requiring many more function calls. Because of the computational cost to calculate the Hessian matrix, especially for costly objective functions, and the relative simplicity of FORM, the Second Order Reliability Method is infrequently preferred.

All methods described up to this point have been so called "Double-Loop Single-Vector" approaches to reliability analysis. In other words, a design decision is made, then a set of analyses are performed at a single design condition to determine the reliability of that design, then the other loop of design decision making is returned to, and so on. There are entire classes of reliability methods which employ a single loop, designing and estimating reliability simultaneously [100, 38, 60, 26]. For reasons which will be covered later, these methods are not considered for the current work. Depending on the results of experiments to be performed, they may be recommended for future work.

Alternatively formulations to probability theory exist including Evidence theory, possibility theory, and interval analysis. These alternatives are ignored for the current work because bounds-based methods like interval analysis and possibility theory (and to a lesser extent, Dempster-Shafer theory) are often bypassed by engineers because they tend to generate more conservative designs relative to probability theory methods [40, 95]. Additionally, Huyse points out that the bounds of engineering uncertainty often cannot be accurately identified [40]. Indeed, sometimes all that can be said with any confidence for an early engineering prediction is that it is 1) probably inaccurate and 2) that the prediction is more likely to be close to the final answer than extremely far off.

A significant problem exists with constraint approximation methods which may limit their applicability to aircraft design problems. Generally, these constraint approximations methods assume that each constraint can be treated independently or that only the most active constraint is relevant to the problem at hand. They, generally, do not consider joint probability distributions [62]. This can be significantly problematic for aircraft design because the selected designs frequently have multiple active constraints, meaning that some form of joint probability is crucial to determining the actual likelihood that a design will be successful under uncertainty.

2.4.3 Probabilistic Methods and Mitigation Actions

Reliability Based Design Optimization is indeed well-suited to ensuring reliability in a design under uncertainty. However, there is no framework established by the literature to include possible mitigation actions taken by the Chief Engineer in an RBDO framework. Choi and Nam demonstrated the closest concept to a mitigation action-equipped framework in the forms of Stochastic Programming with Recourse and Multi-Stage Reliability-Based Design Optimization, respectively. However, both methods have aspects which limit their usefulness for the current problem.

Stochastic Programming with Recourse has limited applicability because the ability to apply infinite recourse to match any other design condition is incongruent with the assumptions at the start of this work - specifically, that mitigation actions are small changes and limited in scope. Additionally, the implication of effectively return to the conceptual design decision during recourse is in direct conflict with the pre-established phases of aircraft design. MSRBDO uses more detailed analyses, requiring more time and additional assumptions on information yet to be obtained. Further, each 'stage' requires both an uncertainty assessment and an optimization loop; this will increase computational requirements exponentially with the number of stages considered. Both Stochastic Programming with Recourse and MSRBDO both fail to account for the fact that the aircraft size and outer mold line will remain fixed after the conceptual design stage.

All existing Reliability-Based Design Optimization methodologies have additional aspects limiting its implementation in practical aircraft designs. RBDO methodologies do not explicitly account for the sophisticated sizing processes typical of aircraft design. RBDO does not account for uncertainty margins of the type typically used in the conceptual design stage of aircraft development.

Again, this work is not intended to change the design process as it stands. The goal is merely to bring information about later stages earlier to allow for better decision making during the conceptual design stage. Thus, it is determined that there exists no directly usable framework for aircraft design including mitigation actions.

The motivation for this thesis remains the same. Thus, it would still be enlightening to consider the possible mitigation strategies that a company's design philosophies would allow. Restructuring an RBDO-like framework to integrate these actions would allow a conceptual design study to mimic more realistic design processes. By examining the impact of the uncertainty space, a probability of compliance can be assessed. Further, by assessing the mitigation space for failed uncertainty scenarios, the process could represent a surrogate of a Chief Engineer's attempts to recover a failed design in the later stages of design.

2.4.3.1 Measuring Mitigation Success

To actually measure the successfulness of mitigation, the process must be examined further. For each outcome of the uncertainty variables, the vehicle performance will be different. Obviously mitigation will not be required if, given an outcome, the vehicle naturally complies with all constraints. Thus, to accurately measure the amount of mitigation required for a vehicle, the method must ensure that for compliant outcomes ($g(x, h, u) \geq 0$) the level of mitigation is zero ($m = 0$).

For non-compliant outcomes the amount of mitigation required will vary depending on which constraints have been violated and by how much. For example, if a design were found to have too high a drag during cruise, it may require some additional fuel, more efficient engines, and/or alterations to the aerodynamics. Conversely, if that same design were instead found to perform adequately in cruise but instead had an excessively long takeoff field length, it would make far more sense to up rate the engine thrust, potentially sacrificing some engine life to meet the takeoff field length performance constraint. Furthermore, the magnitude of each of these problems will dictate the amount of mitigation required in each scenario. This implies that selecting an appropriate level of mitigation for each outcome of the uncertainty variables will not be uniform for all scenarios of a given design. This will incur a "third loop" to the calculation process, where the first two loops are the set of design points and the set of uncertainty points.

2.5 Pareto Optimality

Designing for reliability and performance is a multi-objective optimization with competing objectives. Increasing the probability of success will also tend to increase the weight and fuel requirements of the vehicle. In such a case, a standard optimization routine may be insufficient. While an overall evaluation criterion (OEC) may be used, this formulation requires the decision maker to prescribe preferences between objectives a priori. This may be acceptable in some cases, but the exact tradeoff between these objectives is not yet known. Additionally, a set of designs can be found that are Pareto optimal means that no other design exists which is equivalent or better in all metrics of interest. These designs form what is known as the Pareto Frontier, a set of designs in which to improve in one metric, another must be worsened [28, 80]. Any designs which are not on the Pareto Frontier have at least one other design which is categorically better than it is, meaning it is at least equivalent in all dimensions and is better than it in at least one metric. Thus, only designs lying on the Pareto Frontier will be of interest to a decision maker. Figure 21 shows a example Pareto frontier in two dimensions. By only considering designs lying on the Pareto Frontier, the decision maker can dramatically decrease the number of designs to consider.

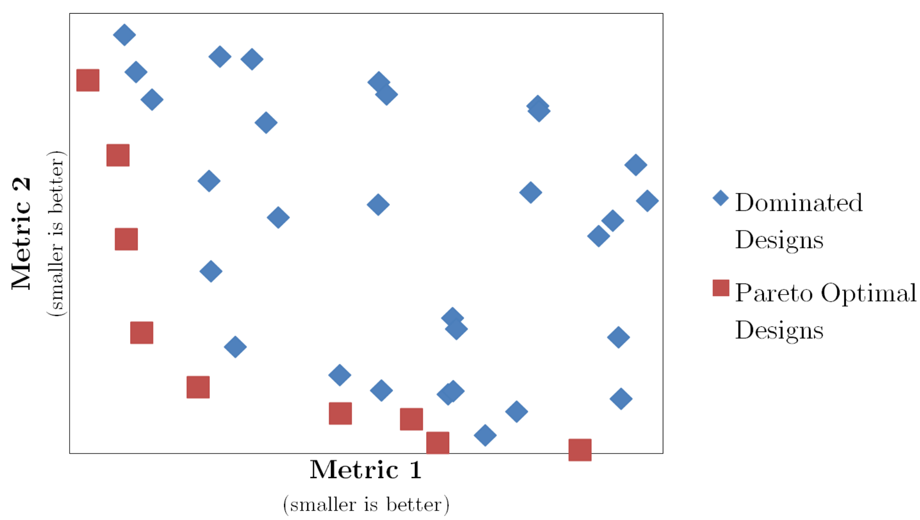


Figure 21: Example Pareto Frontier

The application of Pareto Optimality to RBDO concepts is not without precedent. Zou and Mahadevan used three different approaches to calculate the Pareto Frontier for bi-objective RBDO problem dealing with a car door [99].

If specific probability of compliance and success goals are known during conceptual design, then a single objective, constrained optimization technique will be appropriate. However, should a preference not be specified, it may be prudent to explore Pareto Frontier-finding algorithms.

2.6 Surrogate Models

Surrogate models are a tool for performing analyses more rapidly, rather than a direct method of analyzing design uncertainty. Surrogate models use the principles of regression to create a mathematical model of a more complex code. This model runs much faster than the original, ideally with minimal loss of fidelity. The model is constructed by first establishing a design of experiments, which determines which values to assess using the original tool in order to build the surrogate model [13]. It is important that the range of values used in the design of experiments be large enough to encompass values which may be input into the surrogate model, since the model's predictive power decreases rapidly outside the original data range.

Because of the sheer number of points that must be evaluated in the exploration of mitigation assessments for different uncertainty scenarios across a design space, surrogate modeling will likely be a necessary step of implementing the methodology proposed in this thesis. Of specific value is the reduction of time to run a single loop of the performance analysis. Since the methodology will require running multiple repetitions of the performance tool for different uncertainty and mitigation levels, this reduction will have a great impact on the ability to evaluate the probability of recovery. This will be especially important as the method is tested over the course of its development.

For reliability analyses employing a Most Probable Point of failure, a surrogate model need only be accurate in the region near the MPP, since that it where the reliability is calculated [50]. Allaix and Carbone propose a method to improve the ability of Response Surface Equations (RSE) to accurately predict the Limit State

Function (LSF) [1, 16, 50]. Their method iteratively solves for the Most Probable Point (MPP) of the LSF using the First Order Reliability Method (FORM) on a second order RSE. At the MPP predicted by the RSE, an appropriate rotation of the coordinate system is found for determining new Design of Experiments for the creation of another more accurate RSE. This iterative method shows an improved approximation of the probability of failure. It is unclear whether this method would be amenable to multiple failure modes and thus will not be explored further in this thesis.

Multiple methods exist for implementing surrogate models. Two common ones are explored below due to their popularity, Response Surface Methodology (RSM) [45] and Artificial Neural Networks (ANN) [17].

2.6.1 Response Surface Methodology

Response surfaces are a general class of surrogate models designed to emulate the results of a more detailed analysis code for a smaller subset of problems [61]. They generally take the form of an equation relating the output responses to the set of input variables through a polynomial equation. The generalized equation for a second-order response surface equation is shown in Equation (11) [37].

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \beta_{i,j} x_i x_j \tag{11}$$

Once the model has been created it is tested for accuracy against both the original values and a set of test cases which are independent from the cases to which the model was fit. The metrics used to test accuracy look at how much of the variability of the model is accounted for by the response surface; the distribution of the error in the model to make sure there are no patterns in the errors; the quantiles of the error between the fit cases and the testing cases, as well as the predicted values as compared to the actual input values. If the results of these tests are sufficient the model may be used in lieu of the original analysis code with confidence that the results will be similar.

Due to their polynomial function, Response surface equations are generally less effective with irregularly-shaped response spaces. The next surrogate model implementation examined is one which intends to address this shortfall of RSEs.

2.6.2 Neural Networks

Neural networks, or Artificial Neural Networks (ANN), are a form of surrogate model that work well when the results of an analysis are likely to be non-linear [9, 21, 81]. The principles behind ANNs are modeled on networks of neurons in animal brains, drawing on these systems' ability to recognize patterns and make predictions based on that learning. ANNs are typically constructed with layers of nodes which connect input values to output values. The connections between the nodes and the weights of the connections are updated as the system 'learns' and this process over time results in a surrogate model that is capable of predicting highly nonlinear behavior. The ability to build these models is a common analytical software feature and is therefore widely available.

2.7 Chapter Summary

In this chapter, necessary background information was derived from a review of available literature. The staged aircraft design process was examined. The impacts of uncertainty on this process were illustrated. Previous methods implemented to quantify aircraft design uncertainty were examined for their strengths and weaknesses. Reliability calculation methods employed by RBDO implementations were explored. Pareto optimality for competing objectives was established. A discussion of the use and potential shortcomings of surrogate models was also reviewed.

Research Objective: Quantify uncertainty during the design process for a new aircraft, including a sophisticated sizing analysis, uncertainty margins, and mitigation actions.

Based on this review, gaps still exist within the available literature. There exists no explicit discussion of the treatment of uncertainty when considering both sizing and performance analysis and the dangers of incorrect implementation. Methods exist which treat variables like uncertainty margins, but none exist which tread uncertainty margins directly. There is no existing method which has any implementation mitigation actions. Choi's Stochastic Programming with Recourse constitutes the most closely-related implementation but even that has significant foundational differences with the concept presented in Chapter 1.

These gaps are expounded upon in Chapter 3. The Research Objective is divided into specific research questions. Possible answers to these questions are formulated as hypotheses. These hypotheses are tested in Chapter 4. Based on these results, a step-by-step methodology is proposed in Chapter 5, and the final implementation of this formulation is demonstrated Chapter 6.

CHAPTER III

RESEARCH QUESTIONS AND HYPOTHESES

In this chapter, the information from Chapter 2 is synthesized. This leads to research questions which must be answered in order to formulate the implementation of the methodology. Potential answers are constructed in the form of hypotheses - educated guesses based on the literature and/or preliminary experiments. Once these hypotheses have been developed, experiments can be developed to test the validity of these theories. These experiments are further expounded upon and exercised in Chapter 4.

As stated earlier, the motivation of this thesis is to improve the civil aircraft conceptual design process by integrating information from preliminary design. This indicates that the main goal of this thesis should be to assess the probability of compliance and the probability of recovery for an aircraft design process by incorporating mitigation actions within Reliability-Based Design Optimization frameworks. To this end, a reliability calculation method will be borrowed from RBDO practices. The motivation behind Stochastic Programming with Recourse will be modified to fit the problem formulation at hand. From this, mitigation actions will be incorporated into the reliability framework to find the probability of recovery from accounting for the actions of the chief engineer.

3.1 Implementation of Reliability Analysis

As discussed in Section 2.2, there is inherent uncertainty in a new aircraft design. Modeling this design uncertainty along with its effects is a critical element of this thesis. The effects of this uncertainty will be realized during preliminary design or later in the design process. However, many aspects of the aircraft geometry and design characteristics are frozen at the end of the conceptual design stage. Just as with two-stage stochastic programming, the initial decision variables (x) should not be directly influenced by the uncertainty variables (u) [5]. These frozen characteristics are unable to be altered in later stages of design. Thus, the design selection, geometry, and overall gross weight of the vehicle are fixed before the effects of uncertainty are realized during preliminary design. Instead, other changes - mitigation actions - will be made to the vehicle to try to recover the aircraft in scenarios where it fails to meet one or more performance constraints.

Aircraft sizing tools are not inherently designed to emulate this exact process, specifically with regards to uncertainty and mitigation actions. However, the building blocks necessary to emulate this process do exist. The main building blocks –vehicle sizing and performance analysis– can be employed to create a model of this process. Care must be used with regards to where in the modeling process to apply the uncertainty so that the true design process is emulated accurately. This observation leads to the first research question, below.

Research Question 1 *How should aircraft design with uncertainty be modeled for reliability analysis, accounting for the stages of design?*

The process of modeling aircraft design consists of two steps: vehicle sizing and performance analysis. Because the vehicle will be sized during conceptual design, the vehicle sizing analysis must be used to emulate that stage. Performance analysis will be used to emulate the preliminary design stage because the aircraft geometry and overall weights remain unaltered during both this stage of the design process and this modeling technique. These two stages have a logical order, since the stages of design happen in a chronological sequence. The conceptual design stage occurs before the preliminary design stage; thus, vehicle sizing must occur before the performance analysis.

Fortunately, this structure limits the set of options for where design uncertainty can be modeled. In effect, there are only two competing options: model the uncertainty during the later performance analysis stage or model the uncertainty during the sizing loop. These options are examined in Sections 3.1.1 and 3.1.2, respectively.

3.1.1 Modeling Uncertainty during Analysis

Modeling design uncertainty during the performance analysis means that the uncertainty impacts will be observed only after the vehicle sizing is completed. Thus, the uncertainty scenario(s) will have no impact on the vehicle size or geometry. Restating, a particular design point will be *sized identically*, regardless of uncertainty realization – the same vehicle will always proceed from the sizing step. This vehicle will still *perform* differently due to the influence of the uncertainty variables, as expected of an uncertain design. Logically, modeling this process for a single design would entail the steps depicted in Figure 22 and described below.

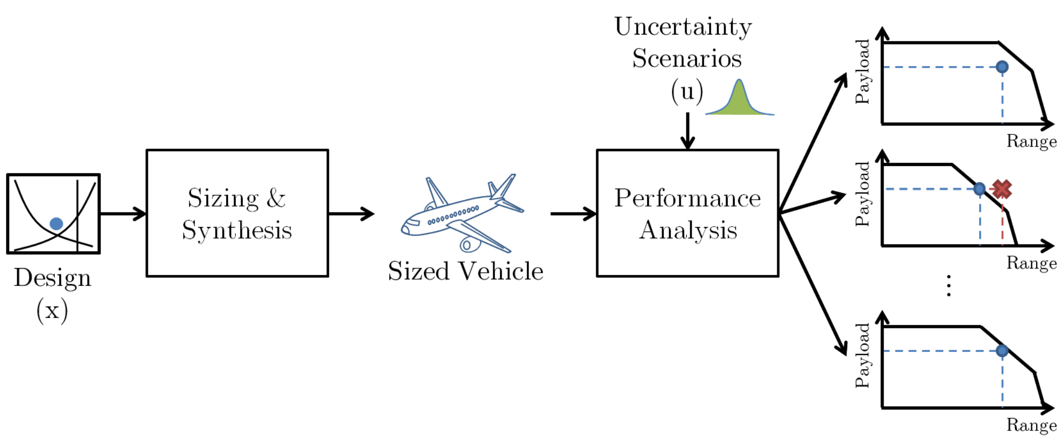


Figure 22: Modeling Uncertainty during Analysis

First, values for design parameters (x) are selected. The design selection may come directly from the designer or via an optimization algorithm. Next the aircraft is sized using those design parameters and a design mission. Additionally, values must be assumed for the uncertainty variables in order to complete the sizing process. The full implications of these assumed values are discussed in Section 3.2. The values of uncertainty variables are introduced after the aircraft sizing is complete. This will change the performance of the vehicle; however, the vehicle's design will not change because sizing has already occurred and design parameters remain unaltered. Finally, the vehicle's performance is analyzed under this new scenario. In response to the changing uncertainty variables, the vehicle's performance (e.g. range, TOFL, etc.) will change.

In other words, this process experiences uncertainty around specific aspects of the design itself. Uncertainty during design stems from the design process mentioned in Section 2.1.1. In conceptual design, rapid tradeoffs are desired so that more designs can be considered and compared quickly. Thus, the models and design tools used in the conceptual phase are faster than those of later phases. In these later phases, higher fidelity tools are used to further refine what is known about the design, reducing the uncertainty in the aircraft's characteristics and performance. However, because these tools are more refined and take more time to run, the overall design is frozen after conceptual design and does not undergo major changes in the later phases.

3.1.2 Modeling Uncertainty during Sizing

Modeling uncertainty in the sizing step of aircraft design is equivalent to changing the design assumption of a vehicle, and creates a different vehicle each time the uncertain value is reassessed. This occurs because the optimizer is allowed to adjust the final design to the specific circumstances associated with the uncertain outcome. For a particular design, this type of process follows these basic steps. First, for each value in a distribution of possible a-priori known uncertainty values different combinations of aircraft parameters are chosen. The aircraft is then sized to those aircraft parameters and uncertainty values. Finally the performance is analyzed at these same values. This process is summarized graphically in Figure 23.

This process experiences uncertainty around the assumptions made during design.

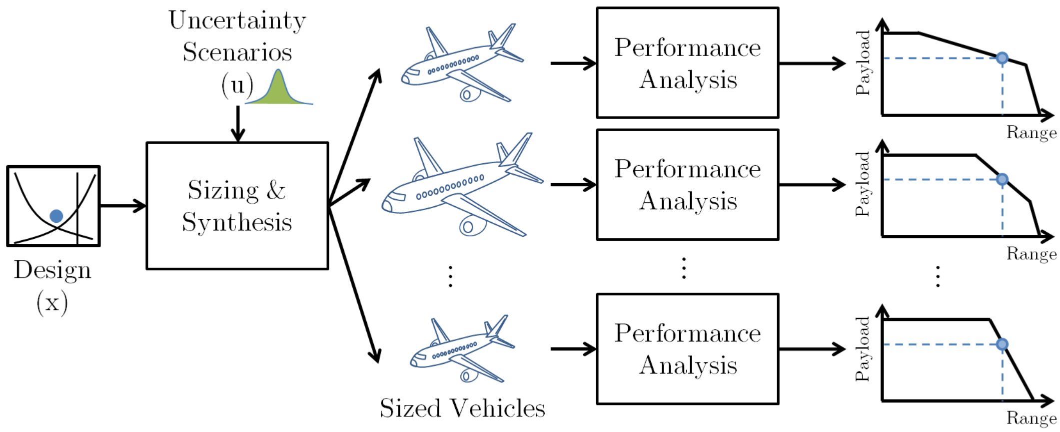


Figure 23: Modeling Uncertainty during Sizing

This type of modeling is most commonly represented by technology studies which exist to assess the usefulness of a suite of technologies on potential future aircraft and to determine which to fund for further development. The end result of this type of study is that a particular suite of technologies will be funded; the aircraft is then designed with these technologies in mind. Assuming the technology development occurs before the actual conceptual design of the aircraft, the technology component impacts (k-factors) will be known at the time of design selection. This means the aircraft sizing step is performed with the uncertain result in mind.

Technology studies are numerous in aircraft design, and many techniques exist to perform this kind of study [14, 45, 70, 85]. Thus, it would be beneficial to use a similar formulation, if possible. To assess the possibility of using this formulation, a comparison must be made to ensure that this method yields similar results to a more directly constructed design uncertainty study.

3.1.3 Uncertainty Implementation Methodology

The act of freezing the design after conceptual design means the aircraft must be sized before the results of uncertain values are determined in preliminary design. Hence it is inappropriate to allow the aircraft sizing process knowledge of the results of the uncertain values. Instead, the sizing should be performed with assumed values of the uncertainty variables, and the uncertainty incorporated during the performance assessment step to determine the effect of the uncertainty on the already sized aircraft.

The process described in Section 3.1.2 does not follow this flow of information. The aircraft sizing process will instead be able to respond to the values of uncertainty variables. This means aircraft will alter to adapt to anticipated values of the uncertainty variables. Furthermore, these guesses will constitute perfect information about the results of these uncertainty variables – information which would not exist during conceptual design. Resizing aircraft based on the uncertainty variables mean that each "design" will produce multiple different aircraft. In other words, while the design vector (x) will be the same, the aircraft will be sized to different weights and therefore will have many differences between each other (e.g. wing areas, thrusts, tail sizes, weights). This is in direct contrast to the problem formulation in which only a single vehicle proceeds into the preliminary design phase. Futher, the problem formulation stated that uncertainty is only assessed once the vehicle configuration has been frozen and has entered the preliminary design phase. Thus, the problem forumlation conflicts with this implementation in two possible ways.

As mentioned, many methods exist which model uncertainty in the sizing analysis. Specifically, technology studies exhibit this behavior frequently [45]. It would be beneficial to make use of these methods now or in the future, if possible. However, the differences in these two formulations also imply that the resulting aircraft may not match. These differences could mean that a reliability analysis performed with the uncertainty resulved during sizing will not give representative answers. This leads to Hypothesis 1.

> **Hypothesis 1** *To emulate the aircraft conceptual and preliminary design process, uncertainty must be implemented after the sizing of the aircraft is complete. Modeling uncertainty during sizing will yield incorrect results for aircraft conceptual design under uncertainty.*

Logically, implementing uncertainty after the sizing analysis will produce a model which gives the desired result of emulating the conceptual and preliminary design processes as described in Section 2.1.1. Due to the potential benefits of using uncertainty implementation before sizing, it will be beneficial to quantify any discrepancies between the two formulations. To this end, Experiment 1 in Section 4.1 will be constructed to test the similarities or differences between these two uncertainty modeling processes.

To represent this process mathematically, Equation (3) from Section 2.1.3.3 is reformulated as Equation (12) to account for probabilistic uncertainty.

$$g_i\left(x,h,u,m\right) \geq 0 := \left(y_i\left(x,h,u,0\right) - y_{req_i}\right) \geq 0 \tag{12}$$

Where $y_i(x,h,u,0)$ is a performance metric evaluated for a specific design, x, an uncertainty scenario, u, and no mitigation. y_{req_i} is the performance limit associated with performance metric y_i, as seen earlier. The terms y_i and y_{req_i} inside the parentheses will reverse for any constraints which limit the upper bound of a performance metric (e.g. takeoff field length).

In this part of the discussion, mitigation actions are not yet considered. However, as will be seen in Section 3.3.2, Equation (12) is just a specific form of the more generalized Equation (13) with the level of mitigation set to zero ($m = 0$).

An outcome of a design is compliant if it satisfies all performance constraints (when no mitigation action is applied), as shown in Equation (13).

$$\left(g_i\left(x,h,u,0\right) \geq 0\right) \forall i \in G \tag{13}$$

Where G is the set of performance metrics and their requirements, $(g_1, g_2, \ldots, g_n)$.

The set of outcomes in which a vehicle is found to be compliant with all performance constraints (A) is defined in Equation (14).

$$(x,h,u) \in A := \left(g_i\left(x,h,u,0\right) \geq 0\right) \forall i \in G \tag{14}$$

Conversely, the set of outcomes in which a vehicle is not compliant with one or more performance constraints ($\overline{A}$) is defined as all combinations of x and u which do not fall in A, specifically show in Equation (15).

$$(x, h, u) \in \overline{A} := (x, h, u) \notin A \tag{15}$$

An indicator function ($1_A(x, h, u)$) can be defined as in Equation (16) which indicates whether a specific combination of settings of design variables and uncertainty variables is included in A, the set of compliant cases

$$1_A(x, h, u) := \begin{cases} 1 & \text{if} \quad (x, h, u) \in A \\ 0 & \text{if} \quad (x, h, u) \notin A \end{cases} \tag{16}$$

The probability that a specific design, x, will be compliant with all performance constraints without the need for mitigation actions ($P(Compliance \mid x, h)$) can be found by using Equation (17). For a design, the integral of the indicator function from Equation (16) is taken over the combinations of uncertainty variables. This result is divided by the total combinations of uncertainty variables.

$$P(Compliance \mid x, h) = \frac{\int_{u \in U} 1_A(x, h, u)\, du}{\int_{u \in U} 1 du} \tag{17}$$

Now, a mathematical formulation has been proposed which will allow a designer to determine the probability of compliance for a given design. Next, the method will be extended to allow for the selection of uncertainty margins to enable finer control over the reliability of the design.

3.2 Modeling Uncertainty Margins

The concept of uncertainty margins was introduced in Section 2.1.1 as a method by which the deterministic design process attempted to account for uncertainty. These uncertainty margins affect the overall performance of the aircraft by forcing the vehicle to size to a different condition. In this manner, uncertainty margins will still be required within a reliability framework because, due to the nature of the sequence of design stages discussed in Section 3.1, the vehicle sizing analysis itself is still deterministic in nature. Section 2.1.1 pointed out that these margins are traditionally determined through some form of heuristic method, be it design experts or previous historically successful values. It would be preferable to instead quantitatively select an uncertainty margin based on a desired probability of compliance. This leads to Research Question 2.

> **Research Question 2** *Is it possible to select a desired probability of compliance and then quantitatively determine a level of margin which will yield that probability of compliance?*

The concept of including a margin in a reliability-based analysis seems counter-intuitive. One of the primary complaints about the deterministic treatment of uncertainty through margins is that said margins will be either optimistic and insufficient for scenarios which could arise. Alternatively, deterministic marings may be overly pessimistic, which will contribute to a more costly design than necessary to ensure the desired level of reliability. However, a margin of some nature may be necessary just to meet basic reliability goals. For example, if the uncertainty requirement was set at 90 percent, then 90 percent of the simulations from sampling the random combinations of uncertainty variables have to be feasible or satisfy all the constraints. Since the nominal values of the uncertainty variables are set at the center of the input distribution, there is a 50-50 chance of either an improvement or degradation from the random variable being selected; thus, achieving the 90 percent feasibility would be virtually impossible. Therefore, a uncertainty margin needs to be introduced as a control variable to remedy this situation. Essentially this uncertainty margin sizes the aircraft for weight which translates to a larger size to account for uncertainty effects. Because uncertainty margins have a large impact on most aircraft design objective function candidates, an optimizer will seek to minimize these variables while maintaining the desired level of reliability. In other words, an optimization routine can be used to determine the appropriate margin for a given design, under the assumed uncertainty distributions.

In theory, if uncertainty margins behave similarly to other design variables, such an implementation should be feasible. To lend evidence toward this assumption, similar implementations were examined further. The concept of using an uncertainty quantification method to set some kind of margin is not without precedent. In 1992 Zhu used reliability methods to determine safety factors for composite aircraft structures and compared them to the traditional 1.5 structural factor of safety [98]. Bristow also combined uncertainty quantification methods with safety target setting [15].

Nam successfully implemented performance target setting in an aircraft design under uncertainty [62]. This performance target setting is conceptually very similar to the idea of uncertainty margins, aside from one key distinction. For performance target setting a constant "push-off" factor is implemented on the performance constraint imposed on the vehicle. This push-off can take the form of an additive or multiplicative factor on the performance constraint of the vehicle, but it will remain a constant level throughout the design space. This idea was shown conceptually in Figure 10 of Section 2.1.1. Motivated by the successful implementation of target setting on an uncertain aircraft design process, it seems reasonable to infer that uncertainty margins can be treated in a similar way.

Uncertainty margins are used to size an aircraft during conceptual design in order for it to be responsive to uncertain conditions. Further, these margins are not forgotten after the aircraft is sized and then frozen; they are treated as actual tools within later design stages. During preliminary design, the Chief Engineer will "allocate" this margin to deal with undesirable changes in the aircraft which arise when previously unknown or uncertain information is refined by more detailed analysis. Thus, the *effect* of the margin will remain as a part of the aircraft while the excess "dead weight" implied by the margin is removed.

Empty weight margin can serve as a conceptually-direct example. If an empty weight margin of 2,000 pounds has been applied to the aircraft during conceptual design, the aircraft will be slightly oversized with respect to the empty weight. The Chief Engineer knows that the empty weight estimate, his design target, contains this "extra" 2,000 pounds above what the conceptual design tools predicted. During these later stages of design, more detailed analyses are performed. After performing detailed structural analyses, each of the individual components of the aircraft could come in at, under, or over their target weight. If the wing is over its target weight, the Chief Engineer knows that there is a repository of weight already baked into the design for just such an occasion. The Chief Engineer can then allocate a portion of this empty weight margin to the wing structures group to keep the entire aircraft at its target weight, despite a component being overweight.

The direct use of uncertainty margins in this manner is straight-forward when considering a single margin and a directly-related uncertainty variable. The concept is demonstrated visually in Figure 24. In this chart, a theoretical (exaggerated) weight breakdown of an aircraft is shown. The y-axis shows the build-up of these weight categories as a percentage of the overall design maximum takeoff weight. Building from the bottom is payload weight in blue, the aircraft empty weight in red, an empty weight uncertainty margin in purple, and the remaining weight available in green. The black line corresponds to the resulting change in design mission range and is related to the secondary y-axis on the right.

The first column in Figure 24 corresponds to the sizing condition of the aircraft. A large margin (exaggerated to ten percent of MTOW) is included in the weight breakdown because the conceptual design codes are assumed to be very imprecise.

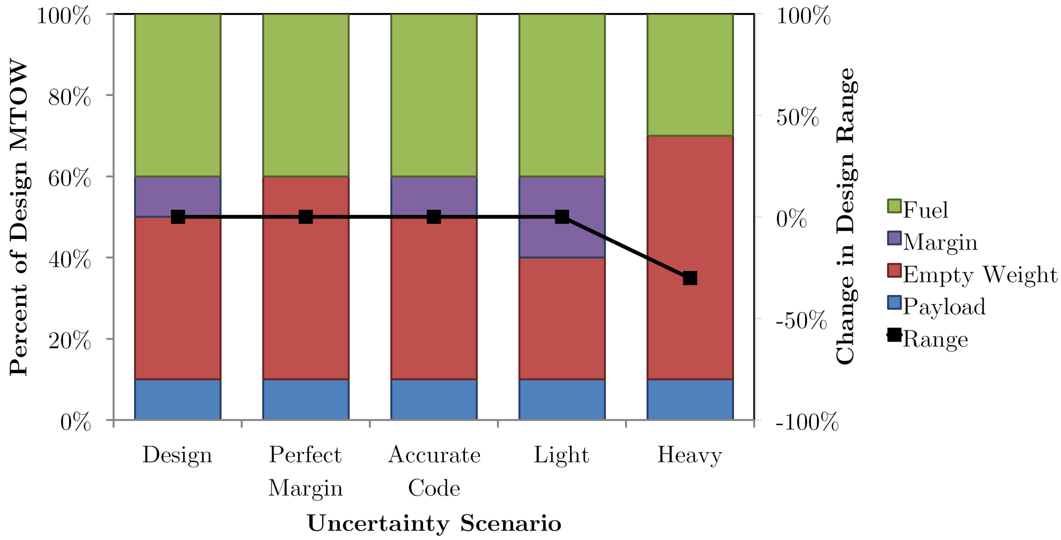


Figure 24: Uncertainty Margin Adapting to Uncertainty Scenarios

The second column shows a "Perfect Margin" scenario in which the final empty weight during preliminary design perfectly matches the predicted empty weight plus the uncertainty margin imposed. This scenario is ideal from a design perspective as the performance of the vehicle will exactly match the anticipated performance. The third column shows the "Accurate Code" scenario in which the conceptual design tool made an accurate prediction of the final empty weight. In this case, all margin from the conceptual sizing remains available for use. Based on the description above, this margin may be maintained throughout all stages of design for other unforeseen circumstances, but it was unnecessary to do so because the weight never increased. The fourth column shows a "Light" scenario under which the final aircraft was actually lighter than predicted by the analysis code. In this scenario, the preliminary design team would actually have extra margin available to respond to future problems which never arise. The final column shows a "Heavy" scenario in which the code prediction plus the uncertainty margin were insufficient to account for the amazingly high weight seen in the later stages of design. Under this scenario the empty weight margin would be unable to keep up with the growth of the vehicle, and the aircraft's design mission range would suffer.

This pound-for-pound offset implementation does not hold when considering multiple margin variables and uncertainty variables simultaneously. Under the implementation as it is described, the aircraft will react to uncertain scenarios to maintain constraint compliance only if the vehicle as designed has remaining uncertainty margin which corresponds *to the particular uncertainty variable which was under predicted*. This behavior implies that a design team would not use a particular uncertainty margin (e.g. drag margin) to maintain the aircraft's compliance with a performance constraint (e.g. design range) if the aircraft were significantly over its target weight. That kind of overly particular decision making would likely get the engineer in charge fired.

It is far more reasonable to assume that all uncertainty margins will be used to their full effect to maintain performance constraint compliance under all uncertainty scenarios. Devising a scheme by which the design team would respond to each uncertainty scenario to maintain aircraft performance would be very cumbersome; a combinatorial examination of uncertainty scenarios for each uncertainty variable and each possible combination of those scenarios and variables would need to be performed to determine the reaction of each uncertainty margin. This exhaustive examination would need to be performed separately for any different combination of uncertainty margin settings. In the end, this process would be incredibly tedious at best and may contain numerous errors at worst.

A far simpler assumption can be made to model uncertainty margins being used to maximum effect to maintain compliance. Instead of a complicated if-then determination of margin responses, uncertainty margins will instead be removed in their entirety when assessing an uncertainty scenario. The concept is demonstrated visually in Figure 25.

In this chart, a theoretical (exaggerated) weight breakdown of an aircraft is shown. The y-axis shows the build-up of these weight categories as a percentage of the overall

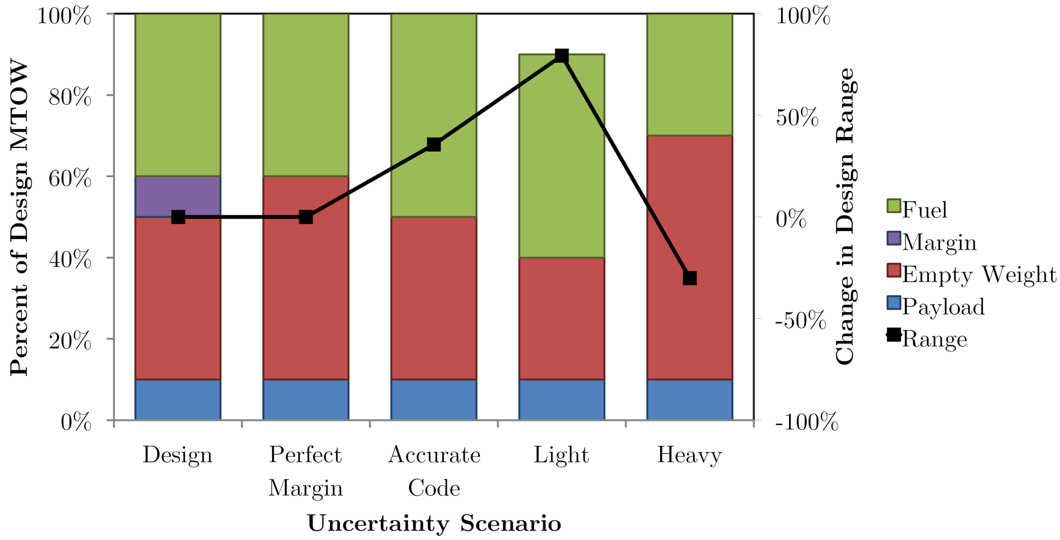


Figure 25: Uncertainty Margin Removed when Assessing Uncertainty Scenarios

design maximum takeoff weight. Building from the bottom is payload weight in blue, the aircraft empty weight in red, an empty weight uncertainty margin in purple, and the remaining weight available for fuel in green. The black line corresponds to the resulting change in design mission range and is related to the secondary y-axis on the right.

Just as with Figure 24, the first column in Figure 25 corresponds to the sizing condition of the aircraft. A large margin (exaggerated to ten percent of MTOW) is included in the weight breakdown because the conceptual design codes are assumed to be very imprecise. The second column shows a "Perfect Margin" scenario in which the final empty weight during preliminary design perfectly matches the predicted empty weight plus the uncertainty margin imposed. This scenario is ideal from a design perspective as the performance of the vehicle will exactly match the anticipated performance. The behavior of Figures 24 and 25 begin to differ at this stage.

The third column shows the "Accurate Code" scenario in which the conceptual design tool made an accurate prediction of the final empty weight. Since the uncertainty margin was removed during the assessment of the uncertainty scenario, there is excess weight available of which the design team can take advantage. Should the fuel tanks contain sufficient capacity, this remaining weight can be filled with additional fuel to maintain the maximum takeoff weight of the aircraft. This additional fuel increases the design mission range of the aircraft. Further, it would allow the unneeded empty weight margin to be used instead to account for detrimental conditions in *other* uncertainty variables like a very fuel inefficient engine.

The fourth column shows a “Light” scenario under which the final aircraft was actually lighter than predicted by the analysis code. In this scenario, the remaining available weight is filled in with fuel up to the point where the aircraft has no remaining space available in the fuel tanks. The design mission range is increased by the addition of fuel to the aircraft and by the forced reduction in weight. This resulting vehicle could also react to detrimental conditions in other uncertainty variables.

The final column shows a “Heavy” scenario in which the code prediction plus the uncertainty margin were insufficient to account for the exceptionally high weight seen in the later stages of design. Just as with Figure 24, under this scenario the empty weight margin would be unable to keep up with the growth of the vehicle, and the aircraft’s design mission range would suffer. However, since other uncertainty margins would be treated similarly to this empty weight margin (i.e. they are removed during uncertainty scenario assessment), these other uncertainty margins may be sufficient to recover the range shortfall.

Including uncertainty margins during the vehicle sizing should be feasible based on previous studies with similar variables. By including these margins during sizing, the designer will have more control over the performance of the aircraft and, therefore, the probability of compliance of the aircraft under uncertainty. Removing the margin before uncertainty assessment offers the distinct advantage that all margins can be used to treat all performance shortfalls, regardless of the offending uncertainty variables, and it does so in a conceptually straight-forward way. The integration of these concepts is formalized in Hypothesis 2.

Hypothesis 2 *By including an uncertainty margin during the sizing process and removing it (but not its effect) before the uncertainty analysis, the impact of margin on the probability of compliance can be seen. Using an optimizer, it will be straight-forward to determine an appropriate level of margin to achieve a desired probability of compliance.*

As a corollary to Hypothesis 2, it should be very possible to extend this method to include multiple designs. Additionally, there is no restriction imposed that would limit other design variables from being introduced into an optimizer solving for margin(s). Thus, an optimization should be feasible which will allow for simultaneous setting of both design variables and uncertainty margins to match a performance goal or a target level of probability of compliance.

It should be noted that this treatment of margins is specific to the uncertainty margins used in this problem. Should other margin types (e.g. performance targets) be used, then this formulation of removing the margin post-sizing may not be the appropriate formulation. Care should be exercised when further developing this methodology or any other methodologies to treat other types of design margins in a reliability analysis.

The mathematical formulation for determining the probability of compliance has been extended to include not only traditional design variables but also uncertainty margins. This will allow for additional control over the probability of compliance for an individual design or the ability to select the best margin for a particular design setting. Next, the designer needs information about the probability of recovery of the failed uncertainty scenarios for that design. This will necessitate the application of mitigation actions to the failed scenarios.

3.3 Modeling Mitigation Actions

From the title of the thesis and many sections before now, it is evident that the formulation of mitigation actions and their implementation will be key to the methodology. Without them, the quantification of design recovery is not attainable. Mitigation plans are currently implemented reactively. As discussed by Sutter, the concept of twisting the wing to alleviate the loads on the 747 was conceived only after a structural problem was discovered [86]. As a concept, this does not aid the goal of bringing knowledge forward into the conceptual design process. However, this does not mean that a manufacturer has never considered possible mitigation actions before. It is likely that a manufacturer has previously established "acceptable" mitigation actions - actions which the manufacturer would prefer not to take but would to recover a design when necessary.

Further, engineers could theorize possible design problems and develop mitigation actions which could be used to address the corresponding performance shortfalls. Just as with reactive recovery during a traditional design process, mitigation actions are formulated to fix specific missed constraints. In order to accomplish this, a process akin to Safety Analysis may be employed. At the most basic level, engineers performing a Safety Analysis will assess different possible accidents that could occur. These accidents are examined to help determine the manners in which such accidents could occur. For each of these different possible problems which could cause an accident, the engineers will determine different possible ways to prevent these specific accidents.

Engineers can develop mitigation actions in a similar manner to Safety Analysis. In order to do so, different possible constraint violations are examined. These violations will be based off the performance constraints imposed on the aircraft design and may be based on regulations (e.g. minimum rate of climb at altitude), access to airports (e.g. maximum wing span), or based on the economics of the aircraft (e.g.

maximum range at a specified payload). Once these possible violations are determined, mitigation actions must be formulated to address these shortfalls. To do so, engineers enumerate the remaining available degrees of freedom in the preliminary design stage. These degrees of freedom are examined to determine which of them can fix the possible performance constraint violations. This step is very important to the overall process herein because without knowledge of the possible mitigation actions which could be taken it is impossible to predict what any unknown actions would do. In order to determine the probability of recovery, a designer needs to define and model possible mitigation actions.

Once these possible mitigation actions are defined, they will need to be assessed. The methodology will need to apply mitigation actions only when they are required. When they are necessary, an assessment will need to be made as to which mitigation actions to employ and by how much to mitigate. Once these mitigation actions are used, their effects on the vehicle needs to be determined.

> **Research Question 3** *How should mitigation be represented in a probabilistic conceptual design model?*

3.3.1 Defining Mitigation Actions

In order to determine the probability of recovery, engineers need to define and model possible mitigation actions. The outcomes of uncertainty variables are discovered after design freeze. Mitigation actions are only implemented in response to performance constraint violations which arise after the outcomes of uncertainty variables are determined. Thus, mitigation actions are only imposed after the design has been frozen. Because of the stage of design in which mitigation actions are implemented, they cannot be large configuration changes. Instead, they must be small changes to fix small constraint violations which can be implemented after a configuration freeze.

Additionally, the late-stage nature of these changes will likely cause detrimental effects which will adversely impact other performance constraints.

Mitigation actions are inherently non-standard design actions. Their application is not codified in a traditional design textbook or course material, nor is it easily found in literature. What can be found is a set of choice examples that clearly illustrate the concepts of mitigation.

Mitigation actions which would be limited to the early phases of design are of little interest here. Since the formulation of the problem specifies that the results of uncertainty are only discovered after the freeze of the aircraft design, any changes which are typically early, large scale decisions like changing the wing planform area will not be considered. Only actions which can be implemented late in the design process after the "design freeze" are of interest.

Mitigation actions are only of interest if they are capable of fixing a problem which could be encountered during preliminary design. Arbitrary changes to the aircraft which do not affect the constraints are not relevant. For instance, since this thesis is focusing on performance constraints, actions which change other metrics like the internal layout of the vehicle or the handling characteristics will not be considered as they do not affect constraints which are being measured.

These mitigation actions are often changes which are non-ideal options. Indeed, their effects could have been executed more efficiently during conceptual design had their need been anticipated. For example, an engine throttle push is, generally, less effective than an engine which was designed to meet the desired thrust condition. Likewise, an aircraft whose wing and pylons were built to handle a specific thrust engine will need to be "beefed up" to handle a higher thrust engine. Mitigation actions will have some form of detrimental penalty on the aircraft, shown conceptually in Figure 26. In this conceptual diagram, a mitigation action is shown on the x-axis which has a direct impact on the overall load of the aircraft (i.e. the maximum takeoff weight). The structural weight of the aircraft is shown on the y-axis. The green dot shows the original design condition to which the vehicle was sized. The red line shows the change in the aircraft as a response to this mitigation action being imposed. Small changes to the maximum takeoff weight are not expected to have a large impact on the structural weight of the vehicle because the aircraft was likely designed with some conservatism. Larger changes to the maximum takeoff weight are expected to increasingly impact the structural weight as more and more reinforcement is required. Eventually, the structural weight penalty will grow so large that the aircraft cannot sustain the changes as indicated by the light blue Recovery Limit line. As a point of comparison, the purple line shows the structural weight of the aircraft if the vehicle were brought back into conceptual design and resized instead of applying mitigation actions.

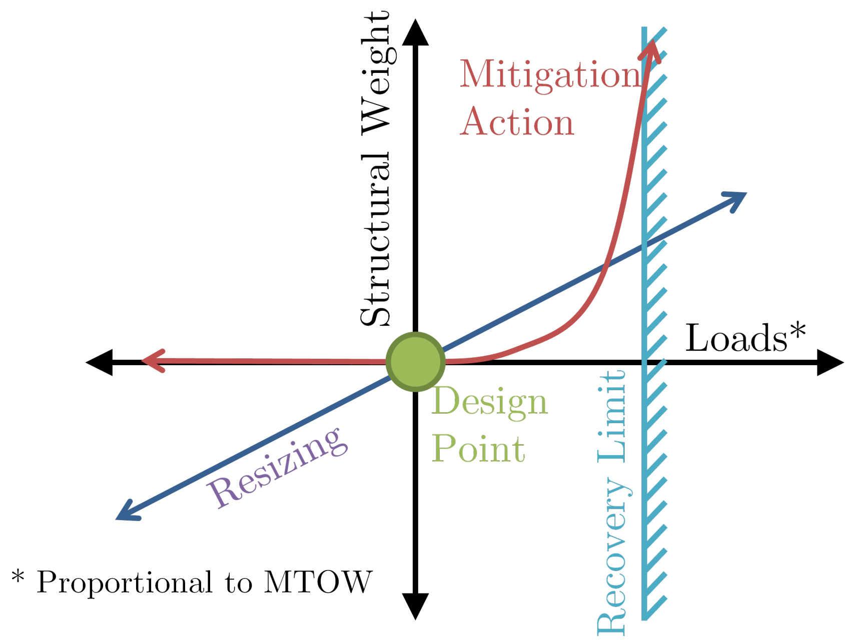


Figure 26: Penalty Associated with a Mitigation Action

The mitigation action will have some direct impact on one or more parameters defining the vehicle with the intent of improving the vehicle performance. In addition to this direct benefit to the vehicle, there is expected to be other detrimental impacts from the application of mitigation actions. These detrimental impacts will be dependent on the changes being made to the vehicle to bring the aircraft back into compliance with performance constraints and may be different for each mitigation action. In the case where multiple mitigation actions are applied, it is expected that all detriments from each mitigation action will be felt on the aircraft.

For the purpose of this work, if the mitigation penalties affect different aspects of the vehicle, both will be applied. However, if the mitigation penalties affect the same parameter, it will be assumed that their impacts are additive. This is shown conceptually in Figure 27. Two different mitigation actions are shown on the x- and y-axes. In addition to their benefit to the vehicle's performance, these mitigation actions enact the same penalty on the aircraft and are conceptually similar to the penalty shown in Figure 26. The penalty associated with any level of mitigation action as well as any combination of the two is shown by the black contour lines. When the contour lines grow closer together, the function value is changing more rapidly.

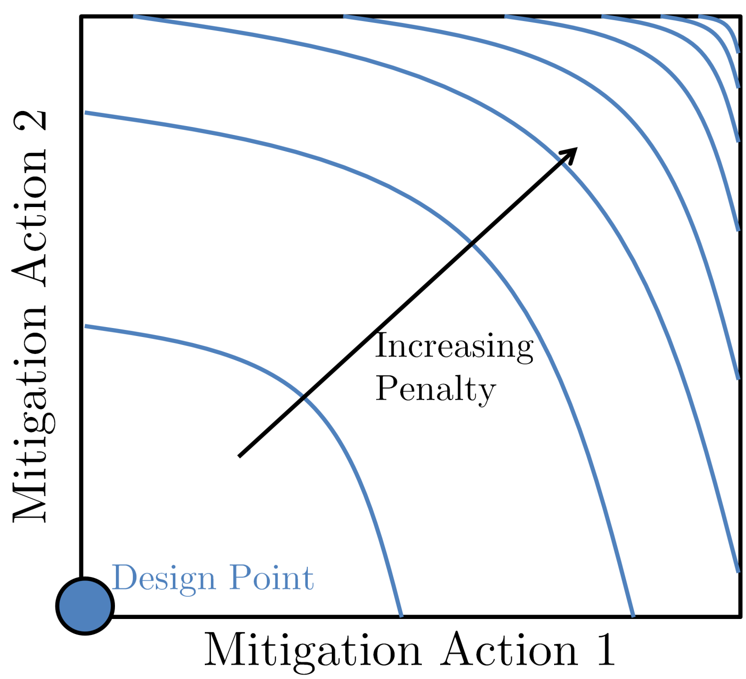


Figure 27: Mitigation Action Penalties are Additive

Mitigation actions should be formulated from a manufacturer's design philosophies, expert judgment from seasoned designers, and basic aerospace engineering concepts. These actions should be constructed in such a way that they are late stage changes. Each action should be designed to remedy an encountered problem. Finally, the implementation of these mitigation actions should capture any realistic detrimental effects on the aircraft.

3.3.2 Representing Mitigation in a Conceptual Model

After the list of possible mitigation actions is developed, the mitigation actions need to be used to quantitatively evaluate the recoverability of the design. Reformulating the constraints from Equation (3) for probabilistic problems and including mitigation actions yields the form seen in Equation (18).

$$g_i\left(x,h,u,m\right) \geq 0 := \left(y_i\left(x,h,u,m\right) - y_{req_i}\right) \geq 0 \tag{18}$$

Where $y_i(x,h,u,m)$ is a performance metric evaluated for a specific design, x, an uncertainty scenario, u, and a mitigation level, m. $y_r eq_i$ is still the performance limit associated with performance metric y_i, as seen earlier.

In the sequence of events being modeled, a mitigation action is only implemented to fix a particular uncertainty condition. Since mitigation actions are an attempt to recover the design's performance rather than face potentially dire consequences due to missing design targets, any acceptable mitigation action would be implemented in that scenario if it would recover the design's shortfall and not violate any other constraints. Thus, an outcome of a design can be recovered through mitigation if there exists any mitigation action in the available set of mitigation actions ($m \in M$) such that all constraints are met simultaneously, satisfying Equation (19).

$$\exists m \in M : \ \left(g_i\left(x,h,u,m\right) \geq 0\right) \forall i \in G \tag{19}$$

The set of outcomes in which a vehicle can be made compliant with all performance constraints through the application of mitigation actions (A_M) is defined as in Equation (20).

$$(x, h, u) \in A_M := \exists m \in M : (g_i (x, h, u, m) \geq 0) \, \forall i \in G \tag{20}$$

The set of outcomes (A_M) in which a vehicle will not be compliant with one or more performance constraints regardless of the mitigation action selected $(\overline{A_M})$ is defined in Equation (21).

$$(x, h, u) \in \overline{A_M} := \nexists m \in M : (x, h, u, m) \in A_M \tag{21}$$

An indicator function $(1_{A_M} (x, h, u))$ which designates whether a design and uncertainty scenario can be recovered is defined in Equation (22). This indicator determines whether a specific combination of settings of design variables and uncertainty variables has any level of mitigation $(m \in M)$ which places it within A_M, the set of cases which are compliant through mitigation.

$$1_{A_M} (x, h, u) := \begin{cases} 1 & \text{if} \quad (x, h, u) \in A_M \\ 0 & \text{if} \quad (x, h, u) \notin A_M \end{cases} \tag{22}$$

The Probability of Recovery $(P (Recovery \mid x, h))$ is defined in Equation (23) as the probability that a specific design, x, can be made to be compliant with all performance metrics through mitigation actions given that uncertainty scenarios are encountered under which the design is non-compliant with one or more performance metrics.

$$P (Recovery \mid x, h) = \frac{\int_{u \in \overline{A}} 1_{A_M} (x, h, u) \, du}{\int_{u \in \overline{A}} (1) \, du} \tag{23}$$

Even for a given design, different outcomes from the uncertainty space will necessitate different levels of mitigation actions. To accurately assess how much of the uncertainty space can be mitigated for a given design, each failed point in the uncertainty space must be assessed to find the appropriate level of mitigation actions to address any missed performance constraints. This process emulates the behavior of the chief engineer in that he or she will fix the outcome that is encountered.

Both the probability of compliance and the probability of recovery through mitigation actions must be considered to determine the overall probability of success of a given design. Some scenarios are compliant without further actions by the preliminary design team. These scenarios were represented by the calculation of $P\left(Compliance \mid x, h\right)$ in Equation (17). Some of the non-compliant scenarios may be brought back into compliance with the constraints through the application of mitigation actions, represented by $P\left(Recovery \mid x, h\right)$ from Equation (23). Therefore, the overall probability of success, $P\left(Success \mid x, h\right)$, is calculated as in Equation (24).

$$\begin{aligned} P\left(Success \mid x, h\right) =& P\left(Compliance \mid x, h\right) \\ &+ P\left(Recovery \mid x, h\right)\left(1 - P\left(Compliance \mid x, h\right)\right) \end{aligned} \tag{24}$$

The expected value of the design objective ($\mathbb{E}[f(x,h)]$) is calculated by summing the design objective values for all compliant scenarios with the design objective of all recovered scenarios and divided by the total number of successful scenarios as in Equation (25). Non-recoverable scenarios are not considered in this analysis because their performance is undefined. The design will be penalized for non-recoverable scenarios through the metric of the overall probability of success ($P\left(Success \mid x, h\right)$).

$$\mathbb{E}[f(x)] = \frac{\int_{u \in A} f\left(x, h, u, 0\right) du + \int_{u \in A_M} f\left(x, h, u, m\right) du}{\int_{u \in A} (1) du + \int_{u \in A_M} (1) du} \tag{25}$$

Since in a real process only one uncertainty scenario would be encountered, the ability for mitigation actions to be applied to each uncertainty scenario independently must be considered. When a scenario causes the aircraft to fail to meet one performance constraint, an appropriate mitigation action will be selected, assuming one exists to fix the violations caused by that scenario. Under a different scenario, a different set of constraint violations may occur. In this case, a completely separate mitigation action may be needed. Between multiple constraints and mitigation actions, it seems reasonable that there may be no overlap between the available mitigation spaces for all uncertainty scenarios. This concept is demonstrated theoretically in Figures 28 and 29.

The left side of Figure 28 shows an uncertainty space constructed using two independent uncertainty variables, u_1 and u_2. Multiple performance constraints are

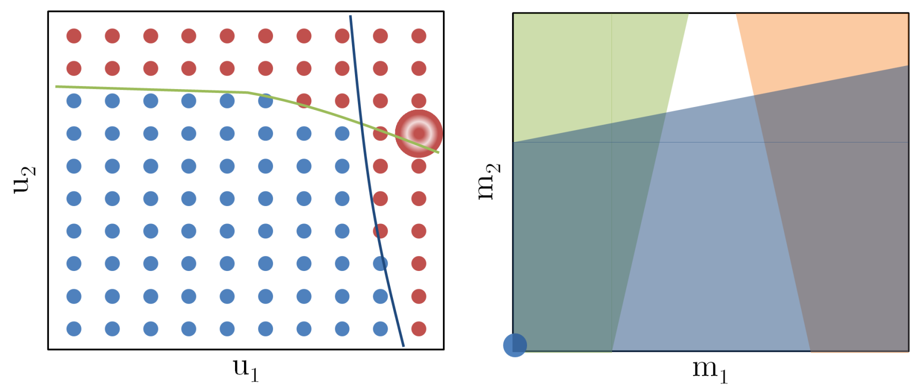


Figure 28: Uncertainty and Mitigation Spaces - Scenario 1

imposed on the aircraft and assessed independently for each uncertainty condition. Whenever all constraints are met, the uncertainty scenario is colored in blue. Whenever the design would fail to meet one or more constraints under that uncertainty scenario, the scenario is colored in red to indicate that it is not compliant. The green and dark blue lines crossing the uncertainty space are the boundary where each of the two constraints becomes active. Within the set of uncertainty scenarios a particular scenario has been selected for further investigation, indicated by the targeted dot on the left. As can be seen, this scenario is violating both the green and blue constraints. The right side of Figure 28 shows the available mitigation space for the design *under this uncertainty condition*. On this side of the plot, two possible mitigation actions are considered, m_1 and m_2, each with a range of applicability. The dot at origin in the lower left corner of this chart indicates that no mitigation action has yet been applied to the vehicle. Colored regions in this space indicate that the aircraft cannot be made compliant by an applied mitigation action with this level of mitigation due to one or more constraints. Just like the uncertainty space, when no mitigation is applied the dot is in both the blue and green colored regions, indicating that the aircraft is violating both of these constraints. A third, orange constraint can be seen; this will limit the range of mitigation actions which can be applied. An available region within the mitigation space is shown in by the white region where no constraint will be violated. According to Equation (20), only one feasible mitigation action which makes the scenario compliant is required to recover a design. Thus, this available space indicates that a possible mitigation action setting exists to recover this uncertainty scenario.

The left side of Figure 29 shows the same uncertainty space as with Figure 28. In this figure, a different uncertainty scenario is selected for further investigation,

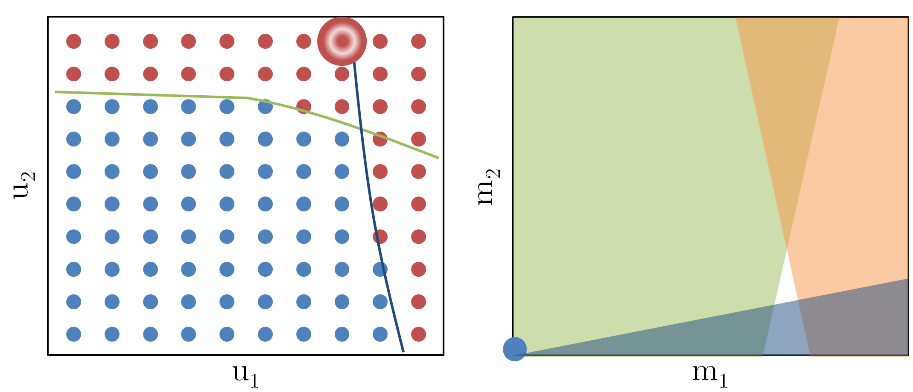


Figure 29: Uncertainty and Mitigation Spaces - Scenario 2

indicated by the red target. For this scenario, only the green constraint has been violated at this time, though the blue constraint is very close to being active. On the right side of the figure, the mitigation space formed between m_1 and m_2 is once again explored. The constraints affect this mitigation space differently than in the previous figure, even though the values which would violate the constraints have not changed. This change occurs because the uncertainty scenario under consideration is different, namely u_1 has decreased while u_2 has increased, changing the performance of the aircraft. Investigating the mitigation space on the right of Figure 29, it can be seen that at the origin on the lower left, only the green constraint is active while the blue constraint is not quite activated. This matches with the observed behavior of the constraints in the uncertainty space on the left. Under this different scenario, a small white region is still available within the uncertainty space, indicating that a set of mitigation actions exist which will bring the design into compliance.

Examining Figures 28 and 29 further can yield some additional insights. The available mitigation region in Figure 29 is very small. This potentially indicates that under more stringent scenarios, the design team may not be able to recover the performance violations through mitigation actions. Thus, the design being considered is likely not infinitely recoverable with only this set of mitigation actions (i.e. $P\left(Recovery \mid x, h\right) < 1$).

Additionally, the mitigation actions necessary to recover Scenario 1 and Scenario 2 clearly have no overlap when comparing Figures 28 and 29. This indicates that there will be no mitigation setting which will recover both scenarios simultaneously. This is acceptable in a real world situation because both scenarios could not be encountered simultaneously for the same design; thus, only one mitigation action setting would need to be applied. However, the realization could complicate the analyses of uncertainty scenarios and mitigation actions. Indeed, if this condition were tested to be true in a representative example, it would indicate that investigating the mitigation space would need to be handled independently for each failed uncertainty scenario.

An alternate possibility would be to attempt to select a single "overall best" level of possible mitigation actions for a given design under all failed uncertainty scenarios. This setting would be the level of mitigation which brings the largest portion of the uncertainty space into compliance. It is theorized that this level of mitigation would yield a lesser probability of success than optimizing mitigation for each uncertainty scenario separately; however, it may reasonably provide a lower bound on the "true" probability of success. Additionally, this restructuring of the problem would remove the need for a MCS approach to assess the reliability, allowing for more sophisticated and faster reliability measures to be employed.

It should be noted that restructuring the problem this way may not buy any time savings unless constraint boundary approximation methods are employed. Approximately the same number of function calls would be needed to optimize the overall best mitigation option because a full MCS reliability measure would be needed at each optimization step. In the original formulation, an optimization would be performed on each MCS instead, leading to the same number of constraint analyses. The exception to this rule would be a surrogate model executed through a code similar to MATLAB, which has the capability to conduct an entire array of analyses much more efficiently than evaluating scenarios individually.

If there were no penalties to the mitigation actions, and all mitigation actions only improved the design, then the maximum level of mitigation could be applied and analyzed. Under these conditions, this maximum level of mitigation would succeed for all uncertainty scenarios which have any possibility of recovery. Unfortunately, both of these conditions are unreasonable. Just like with design variables, some mitigation actions will invariably be opposed to some constraints. As an example, consider the case of a mitigation action to improve the aircraft range. One possible way to recover if this range is violated would be to add additional fuel to the aircraft. Even ignoring any design penalty associated from this extra load, the takeoff weight of the aircraft would need to increase by the amount of fuel added. This increase in weight would cause the aircraft to need a longer field length for takeoff, potentially violating this other constraint as the mitigation action was applied.

Additionally, the need for a penalty function associated with some or all mitigation actions cannot be ignored. The concept is based on the logic that in a real world design process late-stage changes to the design would be imperfect solutions when compared to the alternative of originally setting up the design appropriately. Even if constraints are not in direct opposition to the mitigation action itself, the incorporation of penalties associated with mitigation actions (e.g. additional structural weight for increased loads) means that application of mitigation actions could cause another constraint to become active. Further, the penalty may restrict the application of the mitigation action by itself because the penalty associated with applying increasing mitigation may be more detrimental to the aircraft performance than the mitigation action could offset.

Based on the thought experiments presented here, Hypothesis 3 can be theorized.

> **Hypothesis 3** *It will be necessary to perform a mitigation assessment for each failed uncertainty outcome to get an accurate determination of the probability of recovery.*

The experiment in Section 4.2 will examine the differences between optimizing for each failed uncertainty outcome individually and optimization of the single "overall best" level of mitigation. The experiment will compare the probability of recovery ($P\left(Recovery \mid x, h\right)$) of the resulting aircraft for different design conditions.

3.4 Selection of a Reliability Method

Reliability-Based Design Optimization literature contains a wide variety of methods to calculate reliability of a system, detailed in Section 2.4. Since this methodology is dependent on the reliability of a design as well as characterizing its failure region, one of these reliability frameworks will be employed unless a new one is needed. This thought is summarized in the following research question.

> **Research Question 4** *What reliability assessment method should be used to model the aircraft Conceptual and preliminary design process with mitigation actions?*

If Hypothesis 3 is supported in the experiments, then the ability to assess mitigation actions, not standard probability of compliance assessment, will be the driving factor in selecting a reliability method. As stated in Section 3.3.2, mitigation will need to be implemented differently for each possible uncertainty outcome. Thus, a method must be selected which will allow for the querying of the uncertainty space, rather than a simple likelihood of meeting all constraints. Of the reliability calculation methods reviewed, only Monte Carlo Simulation fits this description.

It is important to note that if Hypothesis 3 is found to be false, it may be possible to use a more efficient method. However, to test this, the related experiment(s) must be established in such a way as to allow for an exact measurement of the recoverability of a design. This setup can then be used as a comparison point or a "truth model." Once this is established, the results of more sophisticated methods can be emulated by establishing constraints on the MCS model to give results similar to the desired method. This can be compared to the truth model to evaluate the amount of information that would be lost by employing such a method. Thus a MCS model must be employed regardless of the results of Experiment 2; it will either be used as a comparison point or as a integral part of the required overall methodology.

3.5 Modeling the Design Process

Methods for handling uncertainty assessment, uncertainty margin selection, and recovery through mitigation actions have been defined. By integrating all of the formulations discussed to this point, a method can be developed by which mitigation actions are incorporated into the uncertain aircraft conceptual design process. This new method will be called Aircraft Recovery through Mitigation & Optimization under Uncertainty for Reliability (ARMOUR) and will be described step-by-step in Chapter 5. ARMOUR will allow for quantification of the effectiveness of mitigation actions and supply that information to a decision maker. The designer will be able to make better decisions about which designs to select, considering both vehicle performance and two reliability measures: the probability of compliance and the overall probability of success.

The core uncertainty quantification and management algorithm of ARMOUR is shown pictorially in Figure 30. The algorithm takes in some setting of the design variables (x) and uncertainty margins (h). This is fed into a Vehicle Sizing algorithm along with a desired design range. After the vehicle is sized, this vehicle is fixed and proceeds on to a Reliability Analysis. The Reliability Analysis takes in the fixed vehicle, distributions on the uncertainty variables (u), and performance constraints (g_{req}) to assess the reliability of the system. Outputs from the Reliability Analysis include the expected performance of the vehicle (e.g. expected economic range block fuel), the probability that the design will be compliant with all constraints simultaneously ($P\left(Compliance \mid x, h\right)$), and the set of uncertainty conditions for which the design fails one or more performance constraints (U_{Failed}). The failed uncertainty conditions for that design are fed into the Mitigation Analysis, along with the performance constraints and the available mitigation actions (m). From this Mitigation Analysis, a probability of recovery ($P\left(Recovery \mid x, h\right)$) will be estimated for that design – the probability that a design can be made compliant through mitigation actions if it fails to meet one or more performance constraints. These probabilities are added together to determine the overall probability of success ($P\left(Success \mid x, h\right)$) for the design. All this information is given to a decision-making algorithm –ostensibly, an optimizer– which can adjust the design variables and uncertainty margins to select a preferred aircraft based off of expected performance characteristics and reliability goals. Once the optimum has been found, the design settings and statistical performance of the selected design are recorded.

Once a design (x) has been selected, the design is fed into a deterministic aircraft sizing tool, designated as "Vehicle Sizing." Design assumptions and a design range

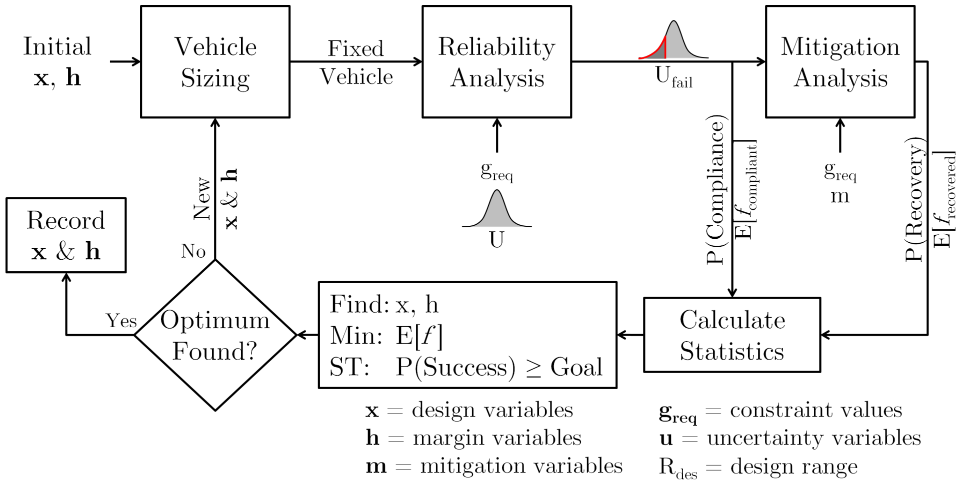


Figure 30: Approach for Conceptual Design under Uncertainty with Mitigation

requirement are also input into the sizing tool. The sizing tool will take this design information and perform a sizing iteration to balance the fuel requirements of the aircraft for that design range with the available fuel. After iterating, the analysis will return a sized vehicle, complemented by detailed aircraft geometry and weights.

The sized vehicle is then "frozen" in terms of the "outer mold line" and overall size, meaning that the Maximum Takeoff Gross Weight ($MTOW$) and the major geometric parameters such as areas, spans, etc. will not change. This fixed vehicle is input into the Reliability Assessment. During this analysis, the uncertainty variables are sampled for different possible scenarios via some sampling method. For each uncertainty scenario, the performance analysis code is executed to evaluate the vehicle's performance metrics. The performance metrics are compared to any applicable constraints, and the vehicle is determined to be either compliant with all constraints or non-compliant. The compliance of the design under this uncertainty scenario is recorded for each sample, and totaled ($u \in A$). Once sampling is complete, the number of compliant cases is divided by the total number of uncertainty scenarios considered to determine the probability of compliance ($P\left(Compliance \mid x, h\right)$) before mitigation actions are applied. For each compliant case, appropriate design objective metric(s) will be returned to the top level optimizer.

$$P\left(Compliance \mid x, h\right) = \frac{\int_{u \in U} 1_A\left(x, h, u\right) du}{\int_{u \in U} 1 du} \tag{17}$$

The failed uncertainty scenarios ($u \notin A$) are then fed into the mitigation analysis. An optimizer is used to find any available settings of mitigation actions which can bring a failed scenario back into compliance with all constraints. If a feasible space is found, the scenario is added to the set of cases which are recoverable ($u \in A_M$), then a minimum acceptable level of mitigation actions is located. At this minimum level of mitigation for the scenario, the design objective metric is recorded for each successfully recovered scenario for use in the expected design objective calculation. If no level of mitigation can be found which brings the scenario into compliance with all constraints, then the scenario is counted as a failure and will not be included in the recoverable designs ($u \notin A_M$).

The probability that a specific design, x, can be made recovered to be compliant with all performance metrics through mitigation actions given that the design is non-compliant with one or more performance metrics ($P\left(Recovery \mid x, h\right)$) is calculated via Equation (23).

$$P\left(Recovery \mid x, h\right) = \frac{\int_{u \in \overline{A}} 1_{A_M}\left(x, h, u\right) du}{\int_{u \in \overline{A}} \left(1\right) du} \tag{23}$$

For every uncertainty scenario where the vehicle is found to be non-compliant, an internal mitigation assessment will determine whether a level of mitigation exists, m_{min}, which will bring the vehicle back into compliance with all performance constraints. If such a level of mitigation can be found, then the scenario is considered recoverable. Also, if $\mathbb{E}[m_{min}]$ is not accounted for during the global optimization, ΔEW need not necessarily be minimized; thus, the "optimizer" merely needs to make the case compliant.

The overall probability that a design will be successful considering the possibility of mitigation ($P\left(Success \mid x, h\right)$) is shown in Equation (24) to be equal to the probability that the design was compliant without mitigation actions plus the probability that the design can be recovered through mitigation actions given that the design was non-compliant multiplied by the probability that the design was non-compliant.

$$\begin{aligned} P\left(Success \mid x,h\right) =& P\left(Compliance \mid x,h\right) \\ &+ P\left(Recovery \mid x,h\right)\left(1 - P\left(Compliance \mid x,h\right)\right) \end{aligned} \tag{24}$$

The expected value of the design objective ($\mathbb{E}[f(x)]$) is calculated by summing the design objective values for all compliant scenarios with the design objective of all recovered scenarios and divided by the total number of successful scenarios as in Equation (25). Non-recoverable scenarios are not considered in this analysis because their performance is undefined. The design will be penalized for non-recoverable scenarios through the metric of the overall probability of success ($P\left(Success \mid x,h\right)$).

$$\mathbb{E}[f(x)] = \frac{\int_{u\in A} f\left(x,h,u,0\right)du + \int_{u\in A_M} f\left(x,h,u,m\right)du}{\int_{u\in A}(1)du + \int_{u\in A_M}(1)du} \tag{25}$$

The probability of success ($P\left(Success \mid x,h\right)$) along with expected value of the design objective ($\mathbb{E}[f]$) is then provided to the designer and his or her optimization algorithm for each design (x). An optimizer can then continue to select a new design vector until it has found a design that yields the best value of the objective function for a required level of probability of compliance and probability of success.

The inclusion of mitigation actions in the modeling of the conceptual and preliminary design process under uncertainty enables the calculation the overall probability of success ($P\left(Success \mid x,h\right)$) of a vehicle. This probability of success includes traditional reliability –the likelihood that a design will be compliant on its own– and the probability that a design can be recovered through late stage aircraft alterations dubbed "mitigation actions." This information is incorporated into a design assessment along with a traditional Reliability-Based Design Optimization objective and set of constraints. Through an optimizer of the designer's choosing, a trade-off can be quantified which allows the decision maker the ability to compare the cost (in terms of traditional design objectives) of increasing overall probability of success of a design.

3.6 Chapter Summary

The beginnings of a methodology have been proposed with the intent of modeling the reliability of an aircraft design process while accounting for uncertainty which is reduced after the conceptual design freeze, the impact of uncertainty margins on this uncertainty, and mitigation actions to recover the design under failed uncertainty scenarios. The sequence of modeling design uncertainty has been established as modeling the uncertainty variables during performance analysis after vehicle sizing is completed. A method for including uncertainty margins in the modeling setup has been proposed with the goal of allowing uncertainty margins to be set in order to control the probability of compliance of a design. Mitigation actions have been incorporated into the methodology to allow for the assessment of the recoverability of failed uncertainty scenarios and to calculate the overall probability of recovery and probability of success of a given design. A sampling method will be used for reliability estimation to allow for the querying of individual uncertainty scenarios in order to accurately determine the probability of recovery. These individual methods are tested in Chapter 4. Based on the conclusions from that chapter, a step-by-step methodology is formulated in Chapter 5 which integrates all of these steps into an overall methodology and incorporates an optimizer. This formulation will allow a designer to select aircraft design points during the conceptual design stage with knowledge of the likelihood that a design will be compliant with all constraints as well as the likelihood that a design will be recoverable should it fail to meet any performance constraints.

CHAPTER IV

HYPOTHESES TESTING

In this chapter, two canonical problems are constructed. The canonical problems are exercised in order to test Hypothesis 1 and 3. Because the two hypotheses address different parts of the overall methodology, the two canonical problems are developed independently.

These experiments were formulated to address the hypotheses devised in Chapter 3. Each experiment is intended to address a hypothesis, which will be repeated during the appropriate section. By evaluating the results of these experiments, the validity of the hypotheses can be tested. The experiments are designed to test the implementation of uncertainty for reliability assessment, the implementation of margins as design variables, and recovery through mitigation actions.

Where possible, a canonical problem will be constructed using design equations available in the collegiate-level literature [2, 75, 78]. Many equations will be necessary to model all the parts of aircraft sizing, uncertainty quantification, and mitigation which are needed for the formulation established in Chapter 3. Instead of a single, large canonical testing apparatus, smaller formulations will be employed to test individual portions of the thesis. The variables used in the experiment will be listed, including applicable ranges and how they will be varied in the experiment. These are designed to allow for maximum transparency into the workings of the proposed method. Since all equations and their arrangement will be described in detail, reconstruction and enhancement of this model by future researchers should be straightforward.

First, a canonical problem to test Hypothesis 1 is demonstrated. This hypothesis deals with the modeling of uncertainty variables in order to emulate conceptual design uncertainty. To test Hypothesis 1 a sizing algorithm is needed which includes equations for Operating Empty Weight (OEW) as a function of Maximum Takeoff Weight (MTOW), mission fuel use equations, and fuel balance equations. Both conceptual design sizing equations and preliminary design performance analysis equations must be implemented in order to emulate the process at hand. Additionally, at least one uncertainty variable is needed, accompanied by affected performance metrics. The resulting canonical problem is exercised to provide the data required to make a determination of the validity of Hypothesis 1.

A second canonical problem is constructed to test Hypothesis 3. This hypothesis focuses on the different mitigation actions necessary to account for different uncertain scenarios. To evaluate the validity of of this hypothesis, performance assessment equations are necessary for each metric of interest. These equations are inherently independent of the sizing algorithm. They must take into account any uncertainty variables as well as any mitigation variables with their associated penalties. Thus, equations are required which account for at least two responses and a mitigation action. Based on the equations selected, two uncertainty variables are used. After accounting for the impact of all variables on the responses, the derived canonical problem is exercised. The resulting data is used to conclude whether Hypothesis 3 is accurate.

These simple test beds will inherently be less inclusive than the final implementation. Their advantage will be in their simplicity and therefore transparency. By using them to test the formulation proposed in Section 3.5, the mechanics of the formulation can be seen directly and will be easy to track for anyone versed in aircraft design.

This information is used to determine the correct implementation of the methodology described in Chapter 5. The precise method arising from the conclusions from these tests is discussed later in Chapter 6.

4.1 Uncertainty Implementation Testing

The examination of the stages of design in Section 2.1.1, the tools used by aircraft designers in Section 2.1.3, and the types of uncertainty experience in aircraft design Section 2.2 led to Research Question 1.

> **Research Question 1** *How should aircraft design with uncertainty be modeled for reliability analysis, accounting for the stages of design?*

After Research Question 1 was posed, a thought experiment was discussed in Section 3.1. Within that section, the impact of modeling design uncertainty in different ways was discussed. The implications were considered for modeling under either competing options: sizing uncertainty and performance uncertainty. The logical argument led to the conclusion that uncertainty needed to be modeled during the performance analysis stage (after sizing), as indicated in Hypothesis 1.

> **Hypothesis 1** *To emulate the aircraft conceptual and preliminary design process, uncertainty must be implemented after the sizing of the aircraft is complete. Modeling uncertainty during sizing will yield incorrect results for aircraft conceptual design under uncertainty.*

However, since many methods already exist which model uncertainty during sizing, a comparison should be made to the processes used by these existing methods to see if their application to this problem would yield incorrect results and to quantify any error.

An experiment is designed to test if the trends of the responses in relation to the uncertainty variables are the same for both methods of uncertainty implementation. Uncertainty variables are varied in a controlled way so that the trends between different implementations can be compared directly. The amount to which they differ will be quantified. If the trends do not compare favorably, the results between the two versions will not be consistent, and the extensively researched implementation of uncertainty during the sizing analysis cannot be used. However, Hypothesis 1 will be refuted if both of the following conditions are met: the two uncertainty implementation methods produce similar trends with respect to the constraints, *and* the probability of compliance between them is found to be comparable.

4.1.1 Process to be Modeled

To answer the question of how design uncertainty should be modeled, one needs to examine the aircraft design process, paying particular attention to when decisions are made and when knowledge is acquired. The aircraft design process described in Section 2.1 breaks the aircraft design into different stages, specifically conceptual design, preliminary design, and detailed design. Conceptual design is when the overall gross parameters of the aircraft are decided. Many parameters are being changed and by relatively large amounts, so the analysis tools used in this stage need to be fast and widely-applicable. Because of these constraints on the design tools, they often have relatively low accuracy, implying high uncertainty. By the end of the conceptual design stage, gross aircraft parameters like wing geometry, engine thrust, and maximum takeoff weight have been selected. In later design stages these settings are frozen and are used as inputs into more detailed tools.

During preliminary design, refinements to the aircraft are made without changing the gross geometric parameters. Additionally, more detailed analyses are used during this stage. Quantities which were unknown or approximated can now be refined through detailed analyses, and their values can be discovered. Thus, the performance of the vehicle will be better understood.

By acquiring this additional information, it may become apparent that the vehicle will fail to meet one or more of the performance requirements. If none are failed, the design will proceed to later stages and manufacturing. However, if a performance requirement is not met, the chief engineer will attempt to address the shortfall and recover performance through any available mitigation actions.

In order to assess the implementation necessary, it will be helpful to examine the process which is being modeled. A graphic of the process to be modeled is shown conceptually in Figure 31. The process sequentially proceeds through conceptual design and preliminary design before transitioning to later stages. In the conceptual design stage, a design point is selected and a sizing analysis performed. Following this stage of design, the design is "frozen," meaning that overall geometric parameters and maximum loads are expected to remain the same, barring extreme difficulties. The vehicle then progresses into preliminary design. In this stage, more detailed analyses are used and more information obtained about the aircraft, leading to refined information. This refined information improves the predictive capabilities of performance assessments. These performance assessments will help the design team learn whether the vehicle is still in compliance with all necessary performance constraints. If the design is not compliant, the design team will attempt to mitigate the design to bring it back into compliance. Following this, the aircraft development will then proceed to later stages of design.

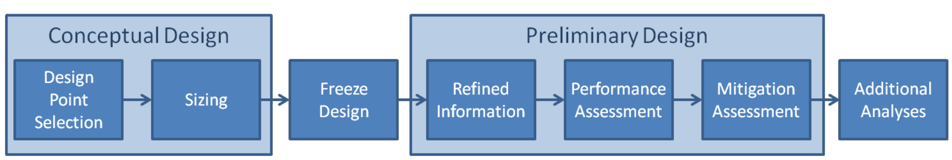


Figure 31: Process to be Modeled

This process has some important characteristics. Conceptual design occurs first. Because conceptual design occurs so early in the design process and simplified tools are often used to make quick decisions, the final quantities of some variables is unknown at this stage. The uncertainty surrounding these quantities will eventually be reduced but at a later date. While this uncertainty is still in effect, the aircraft design must be selected. Once selected, the aircraft will be sized to its final gross weight and physical dimensions. Following this, many of the aircraft parameters will be frozen by the end of the conceptual design stage.

During preliminary design, refinements to the aircraft are made without changing the gross geometric parameters that were previously frozen. More detailed analyses are used during this stage. Quantities which were unknown or approximated are now refined through detailed analyses and their values can now be better determined. As these values change, the performance of the vehicle will be better understood and will likely change to some extent.

It is possible that at this stage the changes in previously uncertain parameters will cause the vehicle performance to fall out of compliance with the performance constraints. If this is the case, the chief engineer will notice any performance shortfalls and react to them using all available tools at his or her disposal. Mitigation actions are an attempt to quantify some of these tools to be able to assess their effectiveness a priori.

4.1.2 Simplified Implementation

Hypothesis 1 can be tested using a canonical problem constructed with aircraft sizing equations from undergraduate aerospace literature. Since this process will occur identically for each design, only a single aircraft design will be considered for observational simplicity. Additionally, only aircraft weights will be evaluated. Thus, only one uncertain parameter will be examined: empty weight uncertainty (u_{EW}). These simplifications will allow for the process to be evaluated both numerically and symbolically, allowing the reasons for any differences to be investigated more thoroughly.

At the most basic level, the process of sizing an aircraft involves the following steps [78]. First a maximum takeoff weight (MTOW or W_0) of the vehicle is guessed. Next, a calculation is performed to estimate the empty weight (W_E) or empty weight ratio ($\frac{W_E}{W_0}$) of the vehicle based on the class of vehicle, the design variables (x), and the maximum takeoff weight. This empty weight is used along with other vehicle weights to determine the resulting vehicle MTOW. These other weights often include the weight of the crew (W_C), the payload including passengers (W_{PL}), and the fuel (W_F). The maximum takeoff weight guess is updated based on this new estimate, usually using some relaxation factor ($0 < \alpha < 1$). An iteration is performed until the input and output maximum takeoff weights are the same. Once the final MTOW is determined, the empty weight ratio can be used to calculate the final empty weight of the resulting vehicle. Rewriting this description as a step-by-step set of directions yields the process below.

1. Guess a takeoff weight (W_{guess})

2. Calculate the empty weight ratio (W_E/W_0)

$$\frac{W_E}{W_0} = f(x) \tag{26}$$

3. Calculate a new takeoff weight (W_0)

$$W_0 = \frac{W_C + W_{PL}}{1 - \frac{W_F}{W_0} - \frac{W_E}{W_0}} \tag{27}$$

4. Update the takeoff weight guess (W_{guess})

$$W_{guess} = \alpha * W_0 + (1 - \alpha) * W_{guess} \tag{28}$$

5. Iterate steps 2-4 until converged

6. Once converged, calculate the vehicle empty weight (W_E)

$$W_E = \frac{W_E}{W_0} * W_0 \tag{29}$$

To proceed with an assessment, an assumed equation will be necessary in place of Equation (26) for step 2 of the process. A representative empty weight equation was pulled from Raymer's undergraduate Aircraft Design book [75]. This equation requires some assumed aircraft parameters. The necessary parameters are shown with their assumed values for a large twin-aisle passenger jet in Table 1. Crew weight is assumed to be negligible for this quick study.

Table 1: Large Twin-Aisle Aircraft Assumptions

Parameter	Variable	Value
Aspect Ratio	AR	8.81
Thrust-to-Weight	T/W	0.3017
Wing Loading	W/S	130.6
Maximum Mach Number	M_max	0.90
Payload Weight (lbs)	$W_{Payload}$	63,210
Fuel Weight Ratio	W_{Fuel}/W_0	0.4372

Using these assumed values, Equation (26) from Step two can be updated. The resulting equation for the empty weight ratio can now be calculated as Equation (30).

2. Calculate the empty weight ratio (W_E/W_0)

$$\frac{W_E}{W_0} = \left(0.32 + 0.92 * W_0^{-0.13}\right) [75] \tag{30}$$

The process described above is obviously a greatly simplified version of the aircraft sizing process. However, it has sufficient granularity to demonstrate the impact of deciding when to implement uncertain parameters in the design process. The following sections will employ this process to implement uncertainty either before or after the aircraft has been sized. These results will be compared to each other and logically compared to the assumed process to be modeled in order to demonstrate when uncertainty should be modeled for this work.

4.1.2.1 Uncertainty Implemented as Sizing Uncertainty

If the results of aircraft uncertainty can be known before sizing, the aircraft can be sized with this information. This process is demonstrated visually in Figure 32. The process shown is constructed based off of Figure 31; however, the uncertainty information is applied during the Sizing analysis in conceptual design. This means that the aircraft dimensions and overall weights will respond to the uncertainty scenario encountered. This assumption is frequently used for future technology funding studies where the aircraft will not enter conceptual design for quite some time.

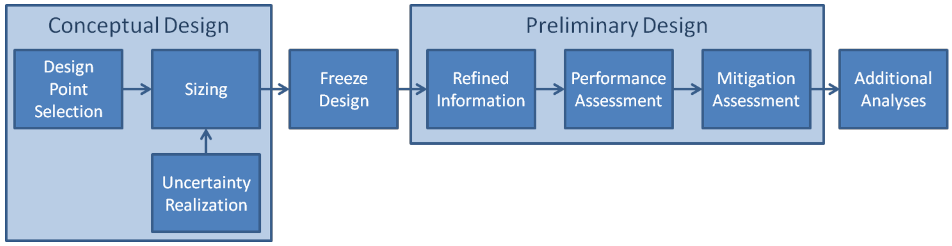


Figure 32: Uncertainty Implemented as Sizing Uncertainty

This would be equivalent to implementing the uncertainty scenarios into the sizing mode of a traditional aircraft design tool. To model this in the simplified process above, the equation for step two, Equation (26), will be modified by including the uncertainty factor, as shown in Equation (31).

2. Calculate the empty weight ratio (W_E/W_0)

$$\frac{W_E}{W_0} = \left(0.32 + 0.92 * W_0^{-0.13}\right) * \boldsymbol{(1 + u_{EW})} \tag{31}$$

This will result in the aircraft size being affected by the uncertainty parameters. Additionally, it should be noted that this is the implied method being used to model uncertainty if the uncertainty parameters are input directly into a standard aircraft sizing tool. By incorporating the uncertainty factor before the vehicle is sized, the maximum takeoff weight (MTOW or W_0) and all resulting parameters (including fuel weight, thrust, and wing area) will adjust to this new value. This means that an increase in empty weight will have effects on most other parts of the aircraft rather than only a direct effect on empty weight. Additionally, the impact of the empty weight factor will be amplified in the iteration loop, leading to further increases in empty weight beyond the specified error.

4.1.2.2 Uncertainty Implemented as Performance Uncertainty

If the values of uncertain parameters are not discovered until after sizing is completed, sizing is not impacted by the uncertainty scenario. This process is illustrated in Figure 33. Again, this process is based off the one shown in Figure 31; however, uncertainty is implemented into the process during the Refined Information stage of preliminary design. Because sizing has been finished before the uncertainty is assessed, this process follows a goal of the design process to be modeled in this thesis: only one design moves forward from conceptual design, regardless of uncertainty condition.

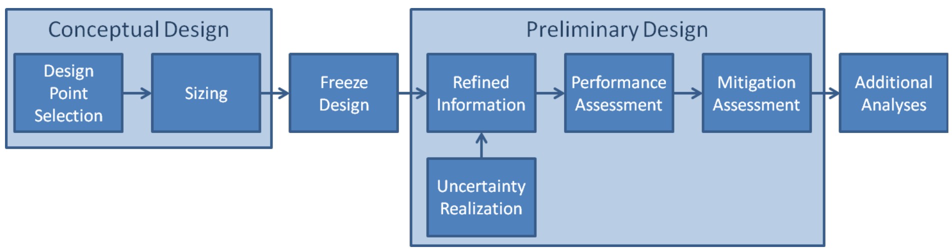


Figure 33: Uncertainty Implemented after Aircraft is Sized

Judging based on Figure 33, the process in Section 4.1.2 will need to be modified by including an uncertainty factor during step 6. This will modify the process above as shown in Equation (32).

6. Once converged, calculated the vehicle empty weight (W_E)

$$W_E = \frac{W_E}{W_0} * W_0 \boldsymbol{* (1 + u_{EW})} \tag{32}$$

By incorporating the uncertainty factor after the vehicle has been sized, only the empty weight changes, while the rest of the vehicle remains fixed. This is consistent with the assumptions of the process to be modeled in Section 2.1. Specifically, the sizing of the aircraft will be completed and the design frozen before uncertainty is implemented.

4.1.3 Uncertainty Implementation Results

Assessing both sets of equations allows from the previous sections allows for a comparison between sizing uncertainty and performance uncertainty. Again, it was determined via logical argument in Section 3.1.3 that performance uncertainty will accurately imitate the design process at the core of this thesis. Since many methods exist which implement sizing uncertainty, it is hoped (but not expected) that this process will yield similar results to allow pre-existing methods to be used.

Figure 34 shows that when uncertainty is implemented after sizing, the Maximum Takeoff Weight of the aircraft does not change. This is consistent with the earlier assumption that the design will be frozen prior to uncertainty being implemented. In other words, the uncertain parameters should not have any impact on MTOW. However, when uncertainty is implemented before the sizing iteration is completed, the uncertain parameter has a large impact on the MTOW of the vehicle, violating prior assumptions. This unwanted effect on the gross weight of the vehicle will be felt in the other assessments. In fact by changing MTOW, the process of implementing uncertainty before sizing is complete has resulted in different aircraft. Because the goal of this study is to assess how an individual aircraft, progressing through the stages of the design process, is affected by the inherent uncertainty in the early stages of design, this result of different vehicles is completely contradictory to the posed problem statement.

For this example, the aircraft design assumption was set at the centroid of the uncertainty space. When implementing sizing uncertainty, the vehicle is sized for the design range in response to the uncertainty condition. Thus, the vehicle will always be compliant with range, regardless of the uncertainty scenario. However, when using performance uncertainty, the vehicle is only compliant with the range requirement 50 percent of the time. It is important to remember that the performance uncertainty is logically correct, as justified leading up to Hypothesis 1. The fact that sizing

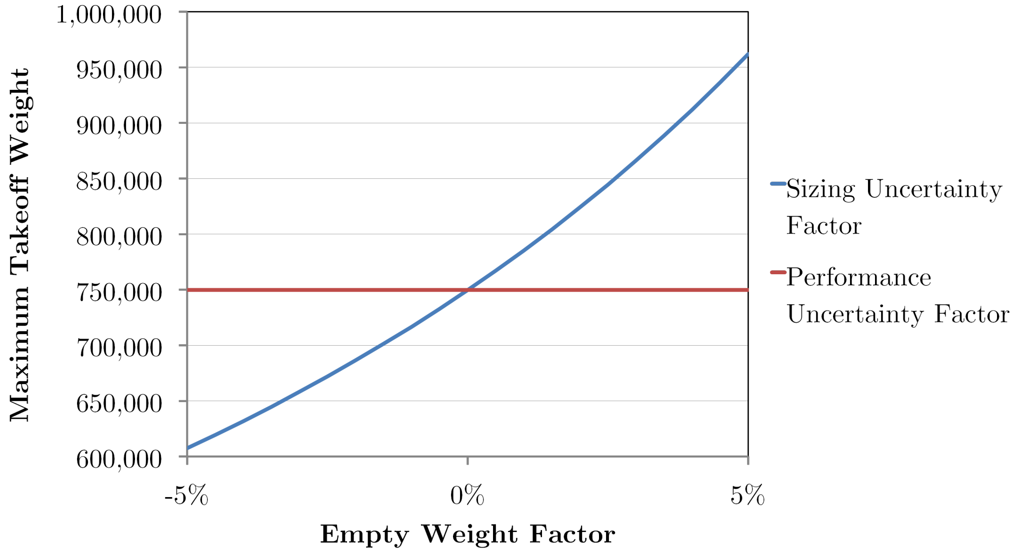


Figure 34: Change in Maximum Takeoff Weight vs. Empty Weight Factor

uncertainty already differs in behavior from performance uncertainty likely means that sizing uncertainty cannot be used for the type of design uncertainty this thesis intends to model.

Figure 35 shows the impact on the Empty Weight of the vehicle by implementing an empty weight factor either before sizing or after sizing. It is very clear to see that the two resulting trends are markedly different. By implementing the empty weight factor before the vehicle is sized, the resulting feedback loop has a dramatic impact on the size of the vehicle. This is due to the sizing loop creating a completely different aircraft than the baseline vehicle. Again, this is inconsistent with the assumption from Section 3.1 that the design will be frozen before the uncertain scenario is realized.

To make a quick estimate of the impact that this difference could have on the vehicle performance, consider the case of landing approach speed. This parameter is proportional to the square root of the landing weight of the vehicle. As the landing weight increases, so does the approach speed. Given that the empty weight is a significant portion of the landing weight of a vehicle, it can be assumed that this change in empty weight will have a large effect on the approach speed. For the sake of a quick analysis, assume that the empty weight accounts for eighty percent of

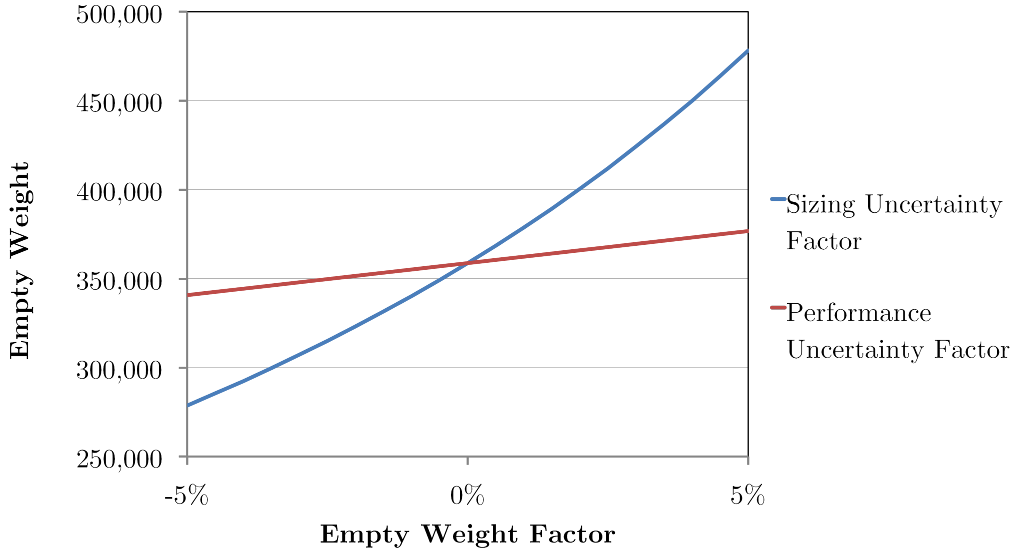


Figure 35: Change in Empty Weight vs. Empty Weight Factor

landing weight and that the other twenty percent will remain constant, regardless of uncertainty condition. The variability of the sizing uncertainty will cause between a 9.4 percent decrease in approach speed and a 12.5 percent increase. If wing loading were instead used as a design variable, then the wing area would adjust based on the maximum takeoff weight seen in Figure 34. This would actually invert the trend of approach speed with respect to empty weight factor, i.e. lower empty weight factors would actually increase the approach speed of the aircraft, albeit by a tiny amount (0.7%). On the other hand, performance uncertainty would have a small variation on the approach speed of +/-2 percent. Because performance uncertainty is only implemented after sizing, this behavior will be the same regardless of whether wing area or wing loading were used as a design variable.

Once again, the trends of fuel available with respect to the empty weight factor are very different depending on when uncertainty is implemented in Figure 36. When the uncertainty is implemented before sizing, the fuel capacity grows alongside the unintended growth in MTOW. This occurs because the aircraft it being sized to meet a given range. Thus, no matter what empty weight factor is implemented, the aircraft will continue to meet its design range. Conversely, if the uncertainty is implemented after sizing is complete, the additional empty weight from the uncertainty factor will eat into the available fuel. This will cause the aircraft to fail to meet range as the empty weight factor increases.

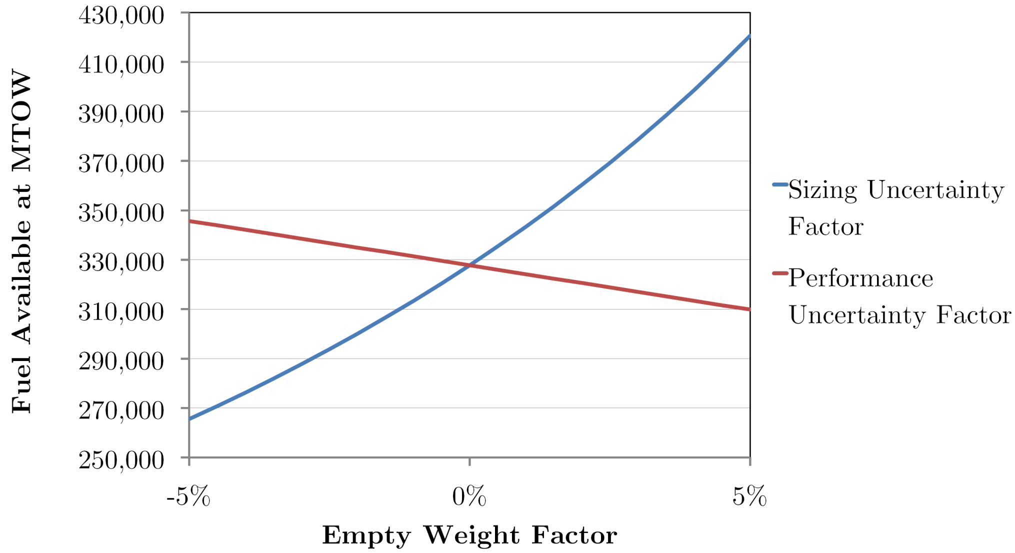


Figure 36: Change in Fuel Available vs. Empty Weight Factor

This established model allows for more that just trend comparison. If an assumed distribution on the empty weight factor is input into the model, the variability of the responses under the two different scenarios can be evaluated. Figure 37 shows the resulting empty weight distributions based off a normally distributed empty weight factor. This figure is the same as Figure 35 except for the addition of the uncertainty distributions. The green distribution along the x-axis shows the input empty weight factor, indicating the frequency under which each of these weights is assumed to occur. Along the y-axis are two additional distributions corresponding to the resulting output distributions of the vehicle empty weight as a percentage of the baseline vehicle empty weight.

It can be seen from the blue distribution in Figure 37 that including the empty weight factor during the sizing loop causes a run-away effect on the resulting empty weight of the vehicle. This is due to the feedback loop inherent in the sizing analysis, and will cause the resulting aircraft to be resized to ensure it meets range. However,

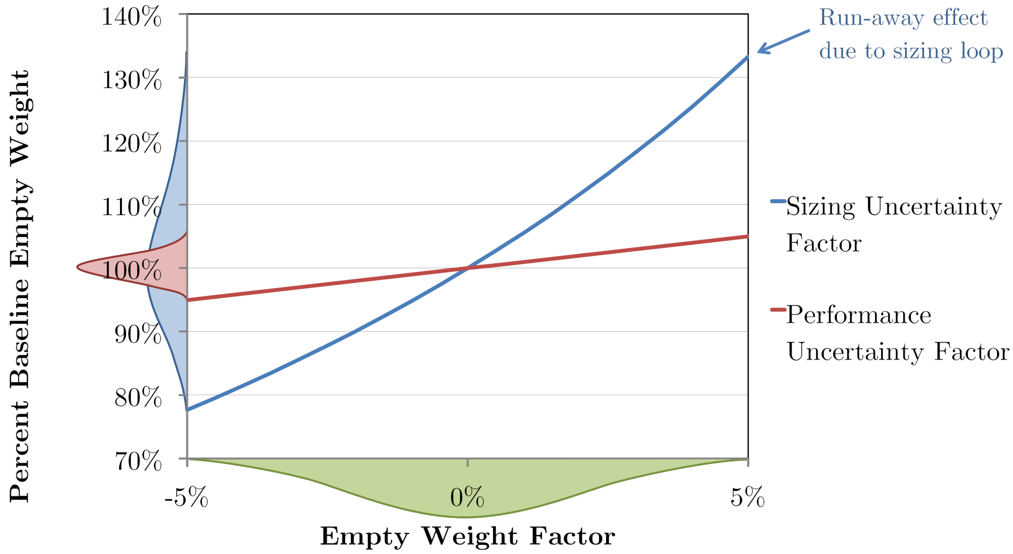


Figure 37: Distributions of Empty Weight Change vs. Empty Weight Factor

this is a false benefit! As discussed in the example of Company Z, the empty weight factor cannot be known at the stage where these decisions are being made. In fact, the true final empty weight of the vehicle is unknown until well after the design is frozen at the end of conceptual design and more detailed analyses can take place. Effectively, placing the empty weight at this stage allows the aircraft to have the *exact* amount of weight margin required, even though such information would not exist. Thus, the resulting blue uncertainty distribution is demonstrably wrong, and using it will give the designer a false sense of security.

The red distribution, which corresponds to implementing the uncertainty during performance, yields a empty weight which varies within about plus or minus five percent of the baseline empty weight. This is consistent with the input empty weight distribution. Other calculations based on this treatment can actually be trusted, unlike the trends resulting from implementing the uncertainty distribution during sizing.

4.1.4 Conclusions Related to Hypothesis 1

The results derived from this canonical indicate that when modeling the uncertainty inherent in a conceptual design, uncertainty must be implemented only after the vehicle has left the sizing loop. This will result in a single, sized aircraft proceeding through later stages of design. The changes in the aircraft due to uncertain factors are appropriately scaled. Further, performance responses to the changes arising from the uncertainty variables will be logically consistent with aircraft performance metrics.

If instead the uncertainty were to be implemented before sizing is complete, as with sizing uncertainty, the resulting responses will be associated with different physical designs. In addition to this logical inconsistency with the posed problem statement, implementing uncertainty before sizing is complete may cause excessive penalties on responses relating to the empty weight of the vehicle (e.g. approach speed). There will also be large penalties on responses which are affected by the maximum takeoff weight of the vehicle like takeoff field length and rate of climb. Additionally concerning is that uncertainty factors would have no impact on the resulting range of the vehicle, which would be very nonsensical.

An important caveat to consider is that this thesis is only making statements about the implementation of the uncertainty during the design process itself. Other types of studies (e.g. technology forecasting) will have different sources of uncertainty, different stages in the design process during which uncertainty is reduced, and correspondingly may have different needs with regards to the process needed to accurately assess those uncertainties. No claims are being made about these other processes aside from specifically conceptual design uncertainty, which is then reduced not only after conceptual design freeze but also before metal is cut.

4.2 Testing the Need for Multiple Mitigation Assessments

The background information within Sections 2.1.2 and 2.2.2 indicated that there was a need to model mitigation actions within a probabilistic aircraft design process. Reviewing Section 2.4.3 indicated that no such method exists within the current literature, leading to Research Question 3.

> **Research Question 3** *How should mitigation be represented in a probabilistic conceptual design model?*

The thought experiments in Section 3.3.2 led to Hypothesis 3

> **Hypothesis 3** *It will be necessary to perform a mitigation assessment for each failed uncertainty outcome to get an accurate determination of the probability of recovery.*

To test this Hypothesis 3, a simplified example problem will be constructed. This simplified problem must be able to assess the performance of a single aircraft under multiple uncertainty conditions. In order to do this, the problem must take into account some performance constraints, and these constraints must be functions of the selected uncertainty variables. Furthermore, it must be possible to see the performance impact on the aircraft due to applying one or more mitigation actions.

In this example, two performance metrics will be considered: the aircraft range at a design payload and the landing approach speed (V_{App}). Range will be assessed using the Breguet range equation [75]. The approach speed will be estimated using the aircraft stall speed and appropriate scalar. These equations will form the backbone of the current analysis.

Two uncertainty variables will be implemented: an empty weight factor (U_W) and a cruise drag factor (U_D). These should allow for sufficient variation while keeping the problem simple. The empty weight factor is expected to affect both range and approach speed performance equations. The drag factor is expected to only impact the Breguet range equation. Any possible impacts from the drag factor on the low speed aerodynamics of the aircraft (and therefore on approach speed) are expected to be negligible.

Only a single mitigation action will be considered: a post-sizing fuel increase with an associated penalty. This mitigation action is intended to improve the aircraft range in the case where this constraint is violated. The addition of extra fuel to the aircraft will directly affect the aircraft's maximum takeoff weight (MTOW) beyond the initial size to which the vehicle was designed. An associated penalty will be imposed on the vehicle empty weight to account for the added structure necessary to carry the extra load. This penalty is expected to increase faster as the additional fuel weight increases; eventually the penalty will restrict the mitigation action's effectiveness at recovering range. Further, this it is expected that the penalty on this mitigation action will have an adverse effect on the aircraft's approach speed performance.

4.2.1 Constraints Imposed

The two performance metrics used in this example problem are the final approach speed of the vehicle during landing (V_{App}) and the aircraft's range for a flight at a fixed payload condition (*Range*).

For the purposes of regulation, the approach speed (V_{App}) of a vehicle, the speed just before landing, is calculated as 1.3 times the stall speed of the airplane [75]. The stall speed can be calculated as show in Equation (33).

$$V_{stall} = \sqrt{\frac{2}{\rho}\frac{W_{landing}}{S}\frac{1}{C_{L_{max}}}}[75] \tag{33}$$

Using this Equation (33) for the stall speed, the approach speed of the aircraft can be calculated using Equation (33).

$$V_{App} = 1.3 * V_{stall} = 1.3\sqrt{\frac{2}{\rho}\frac{W_{landing}}{S}\frac{1}{C_{L_{max}}}} \tag{34}$$

This equation shows that stall speed is a function of the landing weight of the vehicle, its wing area, the local air density, and maximum lift coefficient of the vehicle. For the purposes of this example, it is assumed that the local air density (ρ) is constant. Additionally, the wing area (S) and maximum lift coefficient ($C_{L_{max}}$) will have been set during conceptual design and thus will not change in this example. Only the landing weight ($W_{landing}$) will be affected during this experiment.

The Breguet range equation demonstrated in Equation (35) calculates the distance an aircraft can fly during its cruise segment.

$$R = \frac{V}{C_t}\left(\frac{L}{D}\right)\ln\left(\frac{W_{initial}}{W_{final}}\right) \text{ [75]} \tag{35}$$

The equation assumes that the aircraft will fly at a fixed speed (V) and lift-to-drag ratio $\left(\frac{L}{D}\right)$, optimizing altitude as the weight of the vehicle changes. The fuel consumption (C_t) of the engine is assumed to be constant throughout the cruise segment. The equation is also dependent on the ratio of the initial cruise weight to the final cruise weight $\left(\frac{W_{initial}}{W_{final}}\right)$, indicating how much fuel is consumed during flight.

The mathematics used to generate the following charts distract from the purpose of the discussion and have been omitted. Instead, the discussion here will focus on the main effects and the logic behind the expected behavior. If a detailed examination of the underlying mathematics is desired, the detailed equations employed (Equation (54) through Equation (127)) are located in Appendix A. Two uncertainty variables will be considered in this example: empty weight uncertainty (U_{EW}) and drag uncertainty (U_{Drag}). While the mathematical details are not discussed here, it will be necessary to at least consider the impacts that these uncertainty variables will have on the considered performance metrics.

An uncertainty scenario with high empty weight will result in a decrease in the amount of the vehicle which can be used for fuel. This reduced fuel load at takeoff will reduce the fuel fraction available to be used during cruise. As can be seen in Equation (35), reducing the fuel fraction in cruise will result in a decrease in the vehicle range, degrading its performance. A high empty weight will also increase the landing weight of the vehicle. This increase in landing weight will have a direct impact on Equation (34) and will increase the approach speed of the aircraft, degrading its performance.

Constraints are imposed on both range and the approach speed of the vehicle. These constraints will need to be active for this demonstration because without active constraints all uncertainty scenarios will be considered compliant and will not need any mitigation actions. Since the constraints must be set such that each will be active for at least some part of the uncertainty space, the required levels of performance will be very close to the performance of the aircraft at the centroid of the uncertainty space ($U_{EW} = 0, U_{Drag} = 0$). For this demonstration, the centroid will also be considered to be the design condition of the aircraft. Correspondingly, this vehicle will have a somewhat low probability of compliance, yielding more space to explore for potential mitigation actions. This would be undesirable in a real world scenario, but it is ideal for this test.

All performance characteristics will be normalized about the performance at this centroid. Thus, constraint values close to unity will be selected. Based off some initial testing of the behavior of these equations, constraints were derived. A range constraint of not less than 90 percent of the design range will be imposed on the uncertainty space. Additionally, approach speed will not be allowed to exceed 104 percent of the design approach speed. One mitigation action will be available to recover the aircraft if it should fail to meet the range constraint: a post-sizing fuel addition.

The mitigation action will be a post-sizing fuel addition. This additional fuel will be intended to increase the fuel weight fraction during cruise seen in Equation (35) to increase the resulting range of the vehicle and hopefully bring it back into compliance with the range requirement. This additional fuel will have a detrimental effect on the aircraft. The extra fuel will increase the maximum takeoff weight beyond the weight for which the aircraft was designed, and additional structural weight may be necessary to account for this added load. Thus, an empty weight penalty will be imposed on this mitigation action. This additional empty weight will have a direct impact on the landing weight of the vehicle, degrading the approach speed while the mitigation action attempts to improve range.

Figure 38 shows the uncertainty space associated with these equations, representing a single design. The x-axis shows the empty weight uncertainty variable (U_W) versus the drag uncertainty variable (U_D) on the y-axis. Constraints were imposed on aircraft design range ($\%Range$) and approach speed ($\%Vapp$). The vehicle will violate the range constraint within the blue region. The vehicle will violate the approach speed constraint whenever the uncertainty condition is within the red region. In the purple region, both constraints are violated.

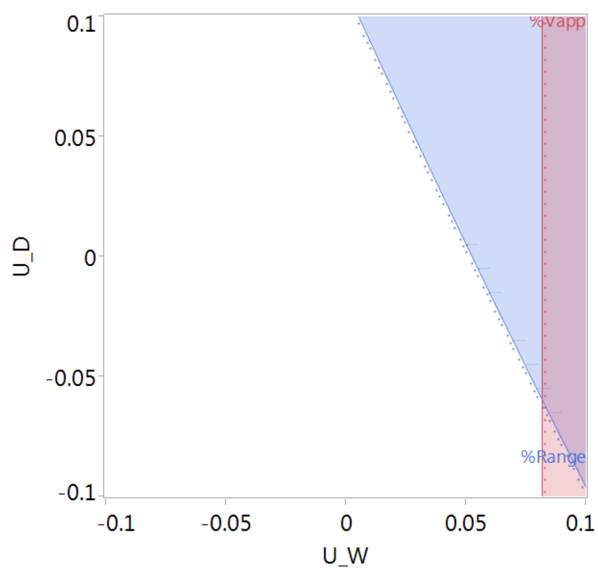
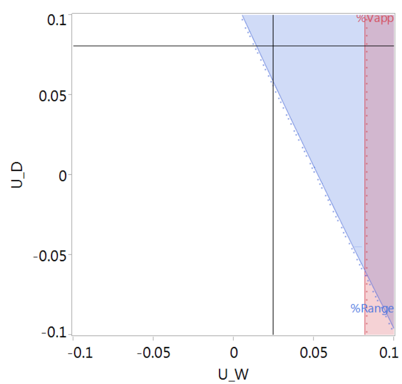


Figure 38: Uncertainty Space for Hypothesis 3 Test

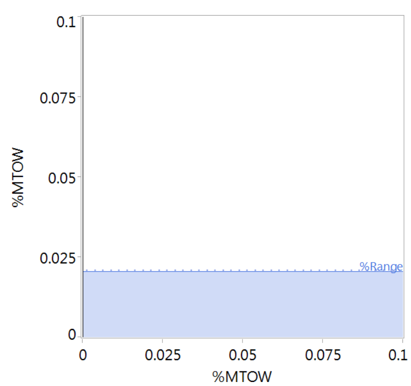
At this stage, no mitigation actions have been imposed. Thus, Figure 38 shows the probability of compliance of the aircraft and as of yet says nothing about the ability to recover the vehicle if it should fail any constraint(s).

4.2.2 Mitigation Assessment Results

Figure 39 shows both the uncertainty and mitigation spaces of the design. On the left, two uncertainty variables –weight and drag– are plotted against each other. The range constraint is violated by any uncertainty scenario on the blue region while the approach speed constraint is violated in the red region. For this set of charts, a specific uncertainty scenario is evaluated at high drag, shown by the circle. The right chart shows the mitigation space available for the design under this uncertain scenario. The mitigation action imposed on the aircraft increases along the y-axis (with the x-axis being a dummy variable). No mitigation action has yet been imposed.



(a) Uncertainty Space

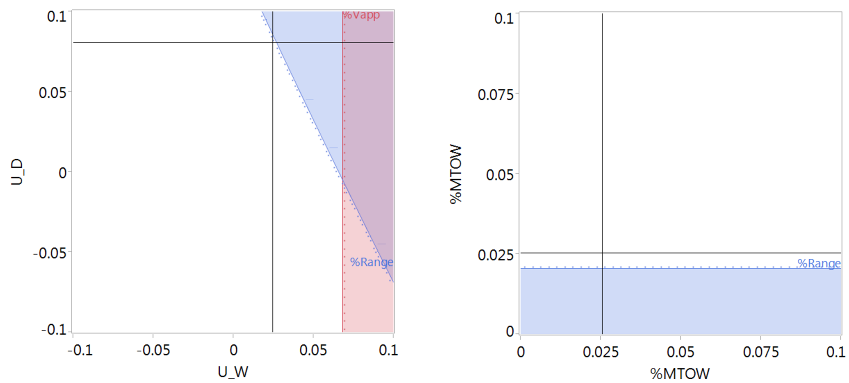


(b) Mitigation Space

Figure 39: Uncertainty and Mitigation Spaces

Looking at the mitigation space on the right of Figure 39, it is apparent that low values of mitigation action will cause the design range constraint to be violated. This fits with the knowledge that the same constraint is violated in the uncertainty space on the left. The position of the constraints within the mitigation space shows that any increase M_{fuel} over about 30 percent of what is available will be sufficient to recover the vehicle and allow it to meet all constraints. Next, the impact of selecting that level of mitigation action is investigated.

Figure 40 shows the implementation of a level of mitigation action determined from Figure 39. Implementing this level of mitigation action in the mitigation space on the right shows that the vehicle should now be compliant with all constraints under the previously selected uncertainty condition. Indeed, after imposing the mitigation action, shown by the circle on the right, an investigation of the uncertainty space on the left demonstrates that both constraints have moved due to the impact of the imposed level of mitigation. The range constraint in blue has been alleviated, allowing for the uncertainty scenario in question to be brought back into compliance. The approach speed constraint in red now covers additional uncertainty scenarios, but since the selected uncertainty scenario was not affected, this condition is considered recoverable through the available mitigation actions.



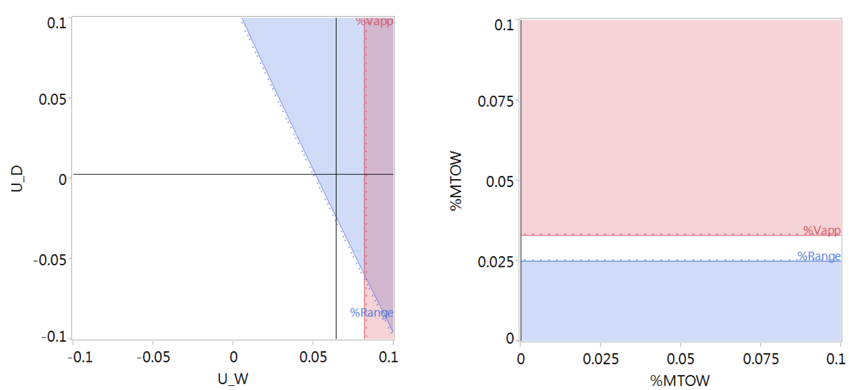
(a) Uncertainty Space (b) Mitigation Space

Figure 40: Uncertainty and Mitigation Spaces with Mitigation Applied

It should be noted from Figure 40 that the mitigation action applied also raised the approach speed of the vehicle, degrading the performance with respect to the V_{app} constraint. This is demonstrated by the V_{app} constraint line moving to encompass more of the uncertainty space. This negative effect was expected based on the equations in Appendix A and the earlier discussion. Since the constraint has not become active for this uncertainty scenario, it is of little direct consequence at this time. However, the behavior should be noted because it will be a concern when considering the mitigation action to apply in other uncertainty scenarios.

This example has shown that the individual uncertainty scenario considered can be recovered through mitigation actions. However, this is insufficient to determine the overall probability of recovery for the entire design. Indeed, each of the uncertainty scenarios which fails to the constraints must be examined to evaluate the probability of recovery. To build up to that, it will be helpful to show the resulting mitigation space when considering a few other uncertainty scenarios.

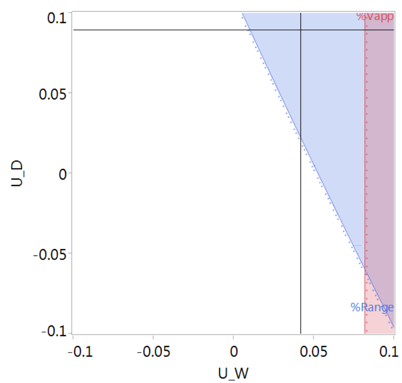
A different uncertainty scenario is selected for Figure 41. In this case, it can be seen on the left side of the chart that the empty weight (U_{EW}) has increased from the design condition but the drag (U_{Drag}) remains at the design condition. This time range is slightly activated about as much as in the original uncertainty scenario. The approach speed constraint is also very close to be activated. On the right side of the chart, the mitigation space can be observed. Once again, range is violated but a small amount of mitigation will recover the design, corresponding to about the same amount of mitigation as in the original scenario. However, unlike the original scenario, the approach speed constraint will become active if too much mitigation is applied, meaning that the scenario will not be recovered. Selecting the specific mitigation action necessary will recover the design and move the constraints in the uncertainty space, but this behavior is not shown for brevity.



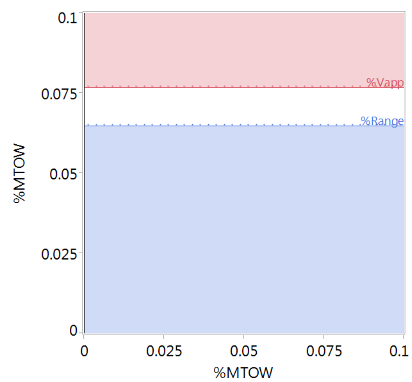
(a) Scenario 2 Uncertainty Space (b) Scenario 2 Mitigation Space

Figure 41: Uncertainty and Mitigation Spaces for Scenario 2

Yet another uncertainty scenario is selected for Figure 42. In this scenario an empty weight (U_{EW}) and drag condition (U_{Drag}) were selected in between the previous scenarios. For this condition, the range constraint is once again violated, this time by a large amount. The approach speed constraint is close to being activated, but not so much so as in Scenario 2. Examining the mitigation space on the right side of the chart reveals that a high level of mitigation is required to recover performance for this uncertainty scenario. Further, the approach speed constraint will activate if too much mitigation is applied, so care must be taken to apply the correct level of mitigation. Selecting this level of mitigation will recover the design and move the constraints in the uncertainty space.



(a) Scenario 3 Uncertainty Space



(b) Scenario 3 Mitigation Space

Figure 42: Uncertainty and Mitigation Spaces for Scenario 3

From the three different scenarios in Figures 40 to 42, distinct behaviors can be seen which are relevant to this work. Even for the same design, different minimum levels of a single mitigation action may need to be applied to recover the vehicle's performance, depending on the uncertainty scenario. Furthermore, a maximum level of mitigation action can be imposed. This prevents the trivial optimization of simply applying the maximum accepted range of the mitigation action to test for recovery.

As mentioned before, to accurately determine the probabilities of recovery and overall success, all the uncertainty conditions must investigated. Doing so allows for the creation of Figure 43, which illustrates whether scenarios were compliant, recovered, or completely failed. The points shown in blue are those which are compliant with all constraints without any need for mitigation action. These uncertainty scenarios contribute to the probability of compliance ($P\left(Compliance \mid x, h\right)$) of the design.

For all the other uncertainty conditions, the mitigation space was investigated independently for each uncertainty scenario to determine if a mitigation action existed which would bring the vehicle back into compliance with the imposed constraints. If a mitigation action could be found which improved the vehicle performance enough to meet all constraints, the uncertainty scenario was then colored green. These green uncertainty scenarios contribute to the probability of recovery ($P\left(Recovery \mid x, h\right)$) of the design. If, while investigating an uncertainty scenario, a mitigation action could not be found which could bring the vehicle's performance into compliance with all of the constraints simultaneously, then this scenario is considered to be failed and is colored in red. These failed conditions indicate an uncertainty scenario in which not only does the vehicle fall out of compliance due to the lack of knowledge present in conceptual design but also that the full set of possible mitigation actions considered to fix a vehicle during preliminary design were insufficient to recover that design's performance to an acceptable level.

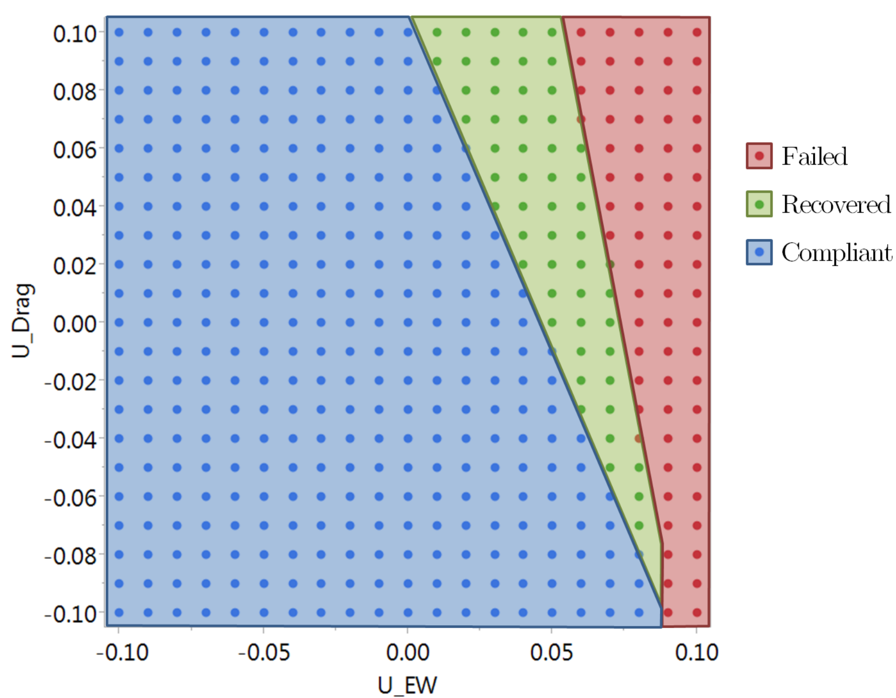


Figure 43: Uncertainty Space Showing Failed, Recovered, and Compliant Scenarios

This simplified problem already shows the necessity for multiple mitigation action settings to determine the probability of recovery. This observation lends weight to Hypothesis 3. Potential interactions like the ones observed here will only increase with the addition of more uncertainty variables, performance constraints, mitigation actions, and penalties on the mitigation actions. Therefore, a method will be needed which appropriately selects a level of mitigation action based not only on the design variables but also on the uncertainty scenario under which constraints were violated.

As a corollary the support for Hypothesis 3 indicates that when considering recovery through mitigation action, some reliability assessment method must be used which allows for querying of the specific scenarios within the uncertainty space. This directly answers Research Question 4 for the proper implementation of this methodology.

4.3 Chapter Summary

In this chapter, two canonical problems were constructed in order to test Hypothesis 1 and 3. Each problem was developed to meet the needs of the individual hypothesis being tested. It was then exercised in order to construct the data required to test the hypothesis. The results of the canonical problems posed in this chapter lend evidence to support Hypothesis 1 and 3. Thus, the implementation of the methodology described in Chapter 5 can be finalized. The details of the resulting implementation is discussed in Chapter 6.

CHAPTER V

FORMULATION

In this chapter, a step-by-step methodology is developed to implement the uncertainty quantification and management algorithm established in Chapter 3. This methodology will bring together uncertainty, reliability requirements, and optimization as commonly found in Reliability-Based Design Optimization (RBDO) methodologies. The methodology will also include the ability to recover design with mitigation actions, and the ability to set appropriate margins to achieve reliability targets will be integrated. Finally, the method will include appropriate treatment of the stages of aircraft design, specifically ensuring that minimal information and analyses are required from later design stages. The method which integrates all of these steps is dubbed the Aircraft Recovery through Mitigation & Optimization under Uncertainty for Reliability (ARMOUR).

Rather than formulating a completely new process, the ARMOUR methodology will build on a commonly accepted aircraft design decision making process shown in the center column of Figure 44, which forms the core of Integrated Product and Process Design (IPPD). This design decision support process can then be augmented to fit the particular needs of the example problem. Its steps assume very little about the design, meaning the process is flexible. This makes it an ideal formulation on which to build the ARMOUR methodology.

The ARMOUR methodology will be broken down into ten steps, as illustrated in Figure 45. Each of these steps corresponds to a particular step of the IPPD decision support process. First, the design team must establish the need for the design as well as the need for the ARMOUR methodology. Next, the team defines

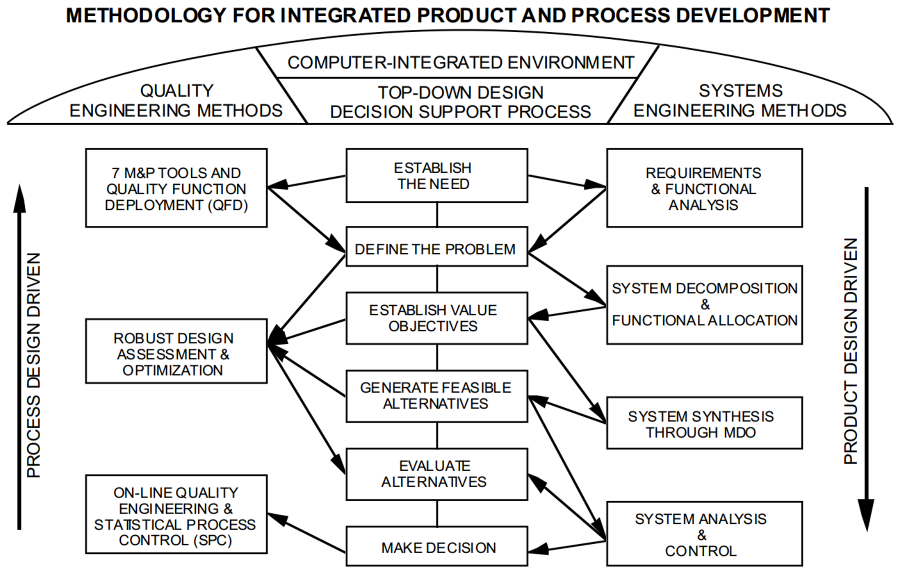


Figure 44: IPPD Methodology at Georgia Tech [55]

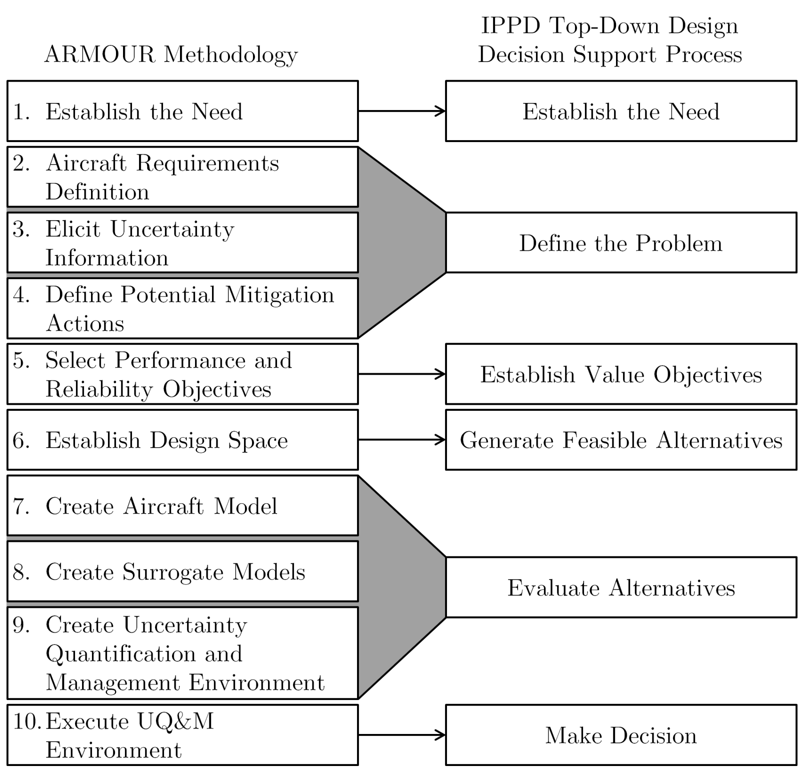


Figure 45: Method Steps Mapped to IPPD Decision Process

the requirements placed on the aircraft, including appropriate performance metrics. Uncertainty information applicable to this design must then be elicited, and potential mitigation actions must next be defined. Next, the team must select performance and reliability objectives, after which an appropriate design space for the aircraft needs to be established. An aircraft model may need to be created, or an existing one may require modification to include all necessary variables. Surrogate models are then constructed from these physics-based models. An uncertainty quantification and management environment is constructed based on the discussion from Chapter 3. Finally, this environment is executed to give information to the designers and help to select an appropriate design. The specific processes for of each of these steps are described in the following sections of this chapter.

5.1 Step 1: Establish the Need

It is assumed that management has decided that a new design or at least a design study is required. A significant process including market research, financial assessments, company design capability assessments, and other factors is often involved in making this portion of the decision; this process is beyond the scope of this thesis. Instead, the need that must be established at this juncture is whether or not the ARMOUR method is required to assess the conceptual design of the new aircraft. Figure 46 shows a flowchart of the questions which must be evaluated in order to determine whether ARMOUR is appropriate.

First, for any reliability-based method to add value, some uncertainty must be present in the design tools. If these tools are perfectly accurate, then any uncertainty assessment method will be unnecessary and a deterministic assessment should instead be used. It is likely that any design team which believes its conceptual design tools to be significantly accurate is naively optimistic.

For ARMOUR to be appropriate, the design process should also be one which will

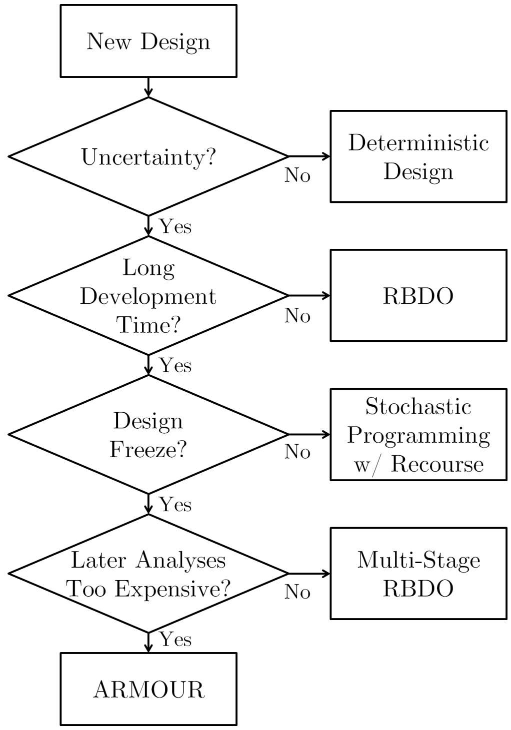


Figure 46: Establishing the Need Flowchart

take significant development time. Specifically, there must be time in the development schedule for some form of corrective action to occur if a performance constraint violation occurs during later design. If instead the design is under a very short development cycle (e.g. component design processes), or if the uncertain results remain unknown until after the design is completely fixed, ARMOUR is not required. In that case, a traditional Reliability-Based Design Optimization method discussed in Section 2.4 would be sufficient.

Some part of the design is expected to be fixed or “frozen” before the uncertainty is fully reduced. This means that the design team will not be able to readjust the full set of design variables during the period in which uncertain results are realized. If a full redesign is possible, even some penalty is associated with doing so, then Stochastic Programming with Recourse discussed in Section 2.3.3 is a more appropriate method.

Finally, it is assumed that later detailed design decisions are too expensive to be sufficiently assessed during the aircraft’s conceptual design. These detailed analyses are often relegated to later stages of the aircraft design process because of the amount of time and personnel required to implement them. Instead, if the required tools are relatively simple to execute and sufficient time exists to run these tools for a large set of design points, then Multi-Stage Reliability-Based Design Optimization discussed in Section 2.3.4 would be a better choice.

If all of these conditions are met, then Aircraft Recovery through Mitigation & Optimization under Uncertainty for Reliability (ARMOUR) is recommended.

5.2 Step 2: Aircraft Requirements Definition

A design architecture should be selected for the aircraft. Additionally, a design mission must be selected for the conceptual aircraft sizing tools. Any additional performance requirements should also be specified along with any constraints which will be imposed on the vehicle. These constraints may take the form of federal guidelines, airport compliance requirements, or point performance objectives. Alternately, these may be internal constraints or desires relayed by potential customers.

5.3 Step 3: Elicit Uncertainty Information

One key aspect at this stage is to determine what uncertainty exists within the current available models and what variables can be used to represent those uncertainties. Since ARMOUR's purpose is to design based on the effects of uncertainty, appropriate variables must be selected to represent uncertain parameters in order to emulate uncertainty in the aircraft design process. Aircraft design principles are examined to determine which variables to use. It is expected that a new design will be subject to a great deal of uncertainty.

Many sources of uncertainty exist during the conceptual design of a new aircraft. It is possible that component weights will be misjudged during conceptual design because insufficient structural analyses are performed that cannot accurately determine the final weights. The assumed vehicle drag characteristics may be inaccurate until the final wing, body, and tail have been developed and all extrusions are accounted for. If the engine is still in development, the aircraft design company may not be fully confident in the engine's performance because the system will be provided by a separate company which they have no direct control over.

These sources of uncertainty can be emulated through many different variables and in different ways within an aircraft model. Obvious candidates to emulate uncertainty include common model tuning parameters like aircraft component weights, vehicle drag polars, engine fuel flow, and low speed lift capabilities.

Additionally, the amount of uncertainty will eventually need to be determined to construct distributions around these variables. This information may not be easy to obtain. In fact, there are entire fields devoted to this subject; one in particular is Structured Expert Judgement. Early on it is most important to determine which variables will be used to model the uncertainty. It is not strictly necessary to obtain the true distributions. The ranges of the uncertainty variables are not required until Step 8: Create Surrogate Models, and the distributions themselves are not needed until the creation of the Uncertainty Quantification & Management Environment in Step 8.

It is also important to consider whether the physics of the problem indicate that the responses selected during Step 2 will be affected by the sources of uncertainty designated herein. If it is determined that the source of uncertainty does not affect any of the responses, then further assessment is warranted. Either an implied response of interest was missed during Step 2 or the source of uncertainty is not relevant to the current problem and should be excluded.

5.4 Step 4: Define Potential Mitigation Actions

Mitigation actions must be constructed such that they can be modeled and analyzed. Acceptable mitigation actions which can be implemented during later design stages should be selected; if this is not the case, these actions will violate the requirement that design is frozen. These mitigation actions should be designed to address possible constraint violations in some manner. The constraints which may be violated can be anticipated based on the aircraft performance responses and the uncertainty effects being modeled.

It is important to remember that the mitigation actions modeled should represent actions which the company's design philosophy will allow. These mitigation actions will likely represent design changes which are not desirable except as a recovery capability to maintain compliance with constraints. Appropriate limits must be determined for the mitigation actions. These limits can be in the form of physical or performance limits on the aircraft or ideological limits imposed by the design team or company.

Secondary effects of the mitigation actions must also be considered at this stage. It is reasonable to assume that any imposed mitigation actions will have some form of detrimental effect on the aircraft. For example, any mitigation action which would impose additional loads on the aircraft would likely increase the stresses on the structure. It would be reasonable to expect that an increase in stresses would require some additional structure, increasing the weight and negatively impacting vehicle performance. It is important to consider such secondary effects or penalties as they may limit the range of applicability of the mitigation action and/or cause other constraints to be violated, a condition which would be very detrimental if missed.

5.5 Step 5: Select Performance and Reliability Objectives

At this point the optimization objective should be determined. This parameter will be dependent on the philosophies and desires of the company or designer making the vehicle selection. Generally speaking, a company will want to maximize the expected potential return on investment of the aircraft or, barring that, to minimize the expected operating cost of the aircraft and make the vehicle more competitive. Many possible objective functions exist within the literature.

Reliability requirements should also be selected at this point. A minimum threshold on the probability of compliance ($P\left(Compliance \mid x, h\right)$) of the design should be specified, as this measure indicates the likelihood that the design will meet all performance constraints without any mitigation and is also functionally equivalent to the reliability measure from RBDO. A goal for the probability of success ($P\left(Success \mid x, h\right)$) should also be defined to indicate the total likelihood that a selected design will meet all performance constraints when accounting for both uncertainty effects and possible mitigation actions.

Additionally, reliability requirements can be imposed for individual constraints as discussed in [60]. Implementing these individual reliabilities is straightforward for the compliance stage of the analysis. If individual constraint reliability goals are desired for the mitigation stage, then the mitigation action implementation may need to be modified. This is discussed at a conceptual level in Section 3.5 but is not implemented in this work.

5.6 Step 6: Establish Design Space

To emulate the many aspects of the problem established in Chapter 3, it is necessary to model a wide variety of variables. Aircraft conceptual design variables are necessary to select a design, since these are the only variables which will be available to a designer. Included in these design variables are uncertainty margins: additional controls to which a designer also has access. Ranges must be established for each of the design variables listed in Step 2.

In order to simulate a new conceptual design of an aircraft, appropriate high-level variables must be selected to change the vehicle's configuration. These variables should have enough influence on the performance of the design that their effects are not completely dominated by the effects of the uncertainty variables. Clearly, if the uncertainty variables are dominant, the design process will be largely irrelevant because it is unlikely that a design will exist which can meet a desired level of reliability. Thus representative variables with a large impact on the design's performance are required.

5.7 Step 7: Create Aircraft Model

As described in Section 6.2, the physics-based aircraft assessment model has specific needs that must be met to properly assess design uncertainty and recovery through mitigation actions. These models are assumed to be aircraft analysis codes appropriate for the conceptual design stage of aircraft development and may have been selected prior to identification of the ARMOUR method for use in this design. The selected toolset must have the ability to model aircraft sizing and aircraft performance as separate conditions. Furthermore, the selection of engine design parameters indicates that an engine analysis tool is necessary. The detailed description of the engine throttle push mitigation action in Section 6.4.3 also indicates a need for an engine assessment capability beyond basic engine design.

As mentioned previously the aircraft analysis code must be capable of both sizing the vehicle and conducting the performance analysis stages described in Section 2.1.3. These stages must be executed in the order described in Hypothesis 1 for this assessment to accurately model the process of aircraft design under uncertainty. The toolset needs to model all desired design variables, accounting for at least gross aircraft-level design variables, some engine-related design variables, some wing-related design variables, and some uncertainty margins during sizing. During performance analysis, the tool must be capable of accepting a frozen aircraft and assessing the performance of that aircraft without resizing while under the effect of different uncertainty variables. Based on the response to Research Question 4 in Section 4.2.2, the tool must also be capable of assessing uncertain scenarios individually. The selected analysis tools need to account for all variables selected in Step 2, 3, 4, and 5. They must also be capable of modeling the sequence of analyses described by Figure 48. The performance analysis must also be capable of including the mitigation actions defined in Section 6.4. Of course whatever model is developed needs to return the performance responses from Section 6.2, so that the impact of all the design, uncertain, and mitigation variables on the performance of the vehicle can be measured.

The toolset needs to model all desired design variables, accounting for at least gross aircraft-level design variables, some engine-related design variables, some wing-related design variables, and some uncertainty margins during sizing. During performance analysis, the tool must be capable of accepting a frozen aircraft and assessing the performance of that aircraft while under the effect of different uncertainty variables. Based on the response to Research Question 4 in Section 4.2.2 the tool must also be capable of assessing uncertain scenarios individually.

Figure 47 illustrates the process which needs to be modeled to generate data useful to the ARMOUR method. First, the model takes in a Design Point: a specific setting of design variables (x) and uncertainty margins (h) around which the aircraft will be sized. The sizing process must compute an aircraft design that matches those parameters and for which the various weight components of the aircraft balance. This defined aircraft, known as a Fixed Vehicle, will now have its performance computed and be input into later analyses.

The fixed vehicle output from the sizing module is frozen, meaning that its dimensions and overall maximum weights can no longer be intentionally changed by the designer or conceptual design code. At this point the uncertainty margins (h) implemented during sizing are removed from the vehicle. The rationalization behind this step is discussed in detail in Section 3.2.

Once the margins have been removed from the fixed vehicle, a specific uncertainty

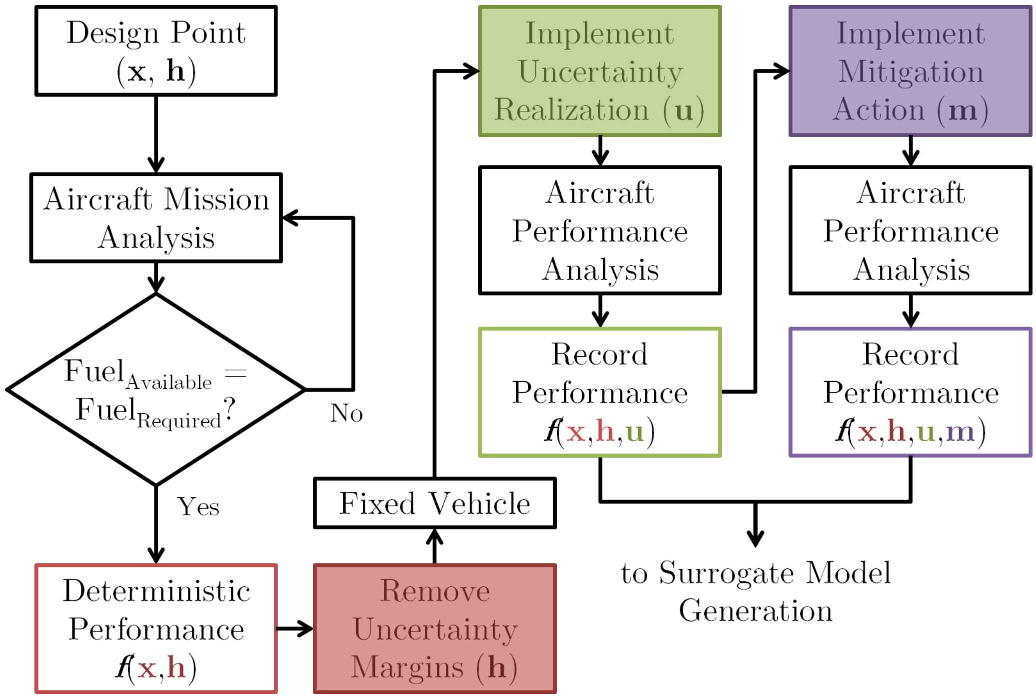


Figure 47: Model Performance Logic

scenario (u) will be assessed. This uncertainty assessment means that the physical dimensions of the frozen vehicle remain fixed, but the non-fixed performance results change according to the specific values associated with the uncertainty scenario. [1] The performance responses from the aircraft analysis are then recorded. These responses will be functions of the design variables (x), the margins (h), and the uncertainty scenario investigated (u).

Next, the Design of Experiments provides a level for each of the mitigation actions (m). These mitigation action settings are implemented on the fixed vehicle model. A fixed aircraft performance analysis is executed to evaluate the response behavior after the mitigation actions have been implemented. Finally, the mitigated performance responses are recorded. These responses are functions of all input parameters to the model, including the mitigation actions and penalties.

[1] Recall that the reason for assessing the uncertainty scenario for the fixed vehicle is that this process simulates a true design process. In the real world, at the point in time when the uncertain design parameters become known, the company has already committed to the physical dimensions of the aircraft.

5.8 Step 8: Create Surrogate Models

The uncertainty quantification & management environment described in Step 9 requires a large number of function calls to assess the probabilities of compliance and success of an individual design. The reliability analysis alone requires so many function calls that multiple methods have been explored to reduce this number at the expense of some accuracy, as discussed in Section 2.4. The addition of the Mitigation Analysis step to assess the probability of recovery of a design will necessitate even more function calls. Even with a fast aircraft analysis code which could evaluate a design in one second, the compliance assessment of an individual design alone could easily take over an hour, making optimization prohibitive. Adding in the mitigation analysis will cause the number of evaluations to grow even larger because multiple assessments will need to be made for each failed uncertainty scenario to determine if mitigation actions can bring the aircraft back into compliance. At one second per function call, the total analysis time to assess the compliance and recovery of a single design could feasibly take over ten hours, eliminating the possibility of any reasonable design optimization.

Due to the large number of function calls needed, using design codes directly will be too time-consuming to obtain useful information for this proof of concept study. Instead, surrogate models of these design tools will be developed. These surrogate models will be input into the uncertainty quantification algorithm developed in Step 9 in place of the sizing and performance analyses. This will alleviate issues related to the speed of the aircraft assessment capability. Surrogates will be created from the data generated by executing the aircraft sizing and analysis tool. The surrogates will emulate performance responses seen in the aircraft design tool as functions of the full set of design, uncertainty, and mitigation variables over a very limited range of application.

In order to create the data needed for the surrogate models, a Design of Experiments (DoE) must be constructed. This will be input into the deterministic aircraft sizing and analysis code described in Step 7. For each case in the Design of Experiments, the deterministic model must be executed and the aircraft performance must be recorded. This performance data along with the DoE will be used as the dataset to generate the surrogate models.

The type of DoE constructed will depend on the type of surrogate models generated. It is recommended that the reader consult Section 2.6 and relevant surrogate modeling methods to determine an appropriate Design of Experiments for his or her problem [9, 17, 21, 37, 45, 61, 81]. Furthermore, each type of surrogate model will potentially necessitate different model generation techniques. It is suggested that the reader investigate the literature on the particular surrogate modeling method desired to determine how to generate that type of model.

5.9 Step 9: Create Uncertainty Quantification & Management Environment

The uncertainty quantification and management environment created here is the crux of the ARMOUR method, so this step is broken into multiple sections. This UQ&M environment assembles the full set of data at the top level. It defines variable ranges, constraints, and mitigation penalties. Probability of compliance and probability of success goals may be defined by the user at this time. After the basics are established, a global optimizer is executed to find an ideal design point, given the defined probability goals and the objective function. For each design assessed by the optimizer, sub processes execute to assess the statistical metrics of probability of compliance, probability of success, and the expected value of the objective function.

The overall algorithm will follow the flow of Figure 48. Before the main uncertainty quantification and management algorithm from Figure 30 is implemented, a set of initial setup steps must be performed. First, all design variables (x), uncertainty margins (h), uncertainty variables (u), and mitigation variables (m) must be defined. The ranges for all of these variables are also defined at this stage, as described in Section 6.6. Additionally, the target design range (R_{Des}) of the aircraft is specified. All of these variables, excluding the design range, are structured into a series of cases called a Design of Experiments (DoE) in Section 6.8. The Design of Experiments will contain values for each of the different variables established in the previous step for each case in the design. These values will inform the aircraft and engine design analyses through the design variables, the uncertainty margins, an uncertainty scenario, and a level of mitigation action to apply. A detailed aircraft and engine code capable of both sizing and performance will execute the DoE cases and determine the resulting aircraft's performance in Section 6.7. Surrogate models of these performance metrics were generated in Step 8 which creates a set of functions relating the performance of the metric to the values of each of the different variables.

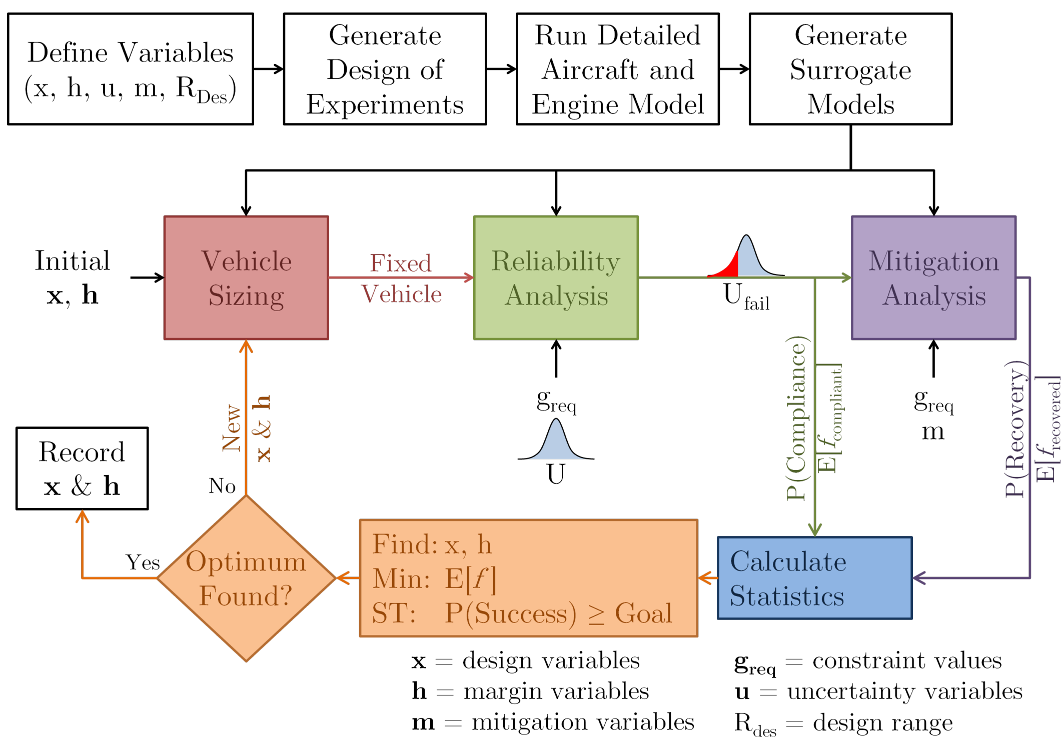


Figure 48: ARMOUR Methodology

Once this initial setup is complete, the algorithm takes in a starting aircraft design (Shown as 'Initial x, h' in Figure 48). This initial design will contain settings for all design variables (x) and uncertainty margins (h) established in Section 6.6. This design point will be input into the aircraft Vehicle Sizing algorithm described in Section 5.9.2. Using the design range (R_{des}) as the sizing condition, the Vehicle Sizing algorithm will create and output a sized Fixed Vehicle corresponding to the input design variable and uncertainty margin settings. Aside from explicitly designated changes due to uncertainty or mitigation variables, this vehicle's design variables must be frozen as-is within later analyses. This requirement has multiple consequences, and they are discussed in Section 5.7.

The now Fixed Vehicle is input into the Reliability Analysis which will assess the overall probability of compliance of the vehicle as detailed in Section 5.9.3. The Reliability Analysis takes in the vehicle performance constraints (g_{req}) and distributions on the uncertainty variables (u). The aircraft performance will then be evaluated for each point in a set of uncertainty scenarios; the uncertainty scenarios will be generated by using a sampling method to draw from the input distributions of the uncertainty variables. The performance of the Fixed Vehicle under each uncertainty scenario is then compared to the performance constraints; under each scenario the Fixed Vehicle will either meet all constraints or fail to meet one or more constraints. This dataset is then used to assess the probability of compliance ($P\left(Compliance \mid x, h\right)$) of the vehicle as well as the expected value of the objective function ($\mathbb{E}[f(x, h)]$). This data will later be processed by the optimizer. The uncertainty scenarios which cause the Fixed Vehicle to fail to meet one or more performance constraints are stored as failed uncertainty scenarios (U_{Failed}) to be input into the Mitigation Analysis.

The Mitigation Analysis described in Section 5.9.4 determines the probability of recovery ($P\left(Recovery \mid x, h\right)$) of the Fixed Vehicle. The failed uncertainty scenarios (U_{Failed}) from the Reliability Analysis are input into the Mitigation Analysis along with the performance constraints (g_{req}) and the possible mitigation actions (m). This Mitigation Analysis will assess the recovery of the Fixed Vehicle under each of the failed uncertainty scenarios (U_{Failed}) by attempting to find any combination of mitigation actions which will bring the design back into compliance for that scenario, indicated by Equation (19). By assessing all failed uncertainty scenarios, the probability of recovery ($P\left(Recovery \mid x, h\right)$) of the Fixed Vehicle can be calculated and is output from the Mitigation Analysis.

The outputs of the Reliability Analysis and the Mitigation Analysis are brought together to calculate the overall probability of success ($P\left(Success \mid x, h\right)$) of the design. This metric represents the likelihood that the aircraft design (x and h) will be compliant with all performance constraints (g_{req}) given the distributions of the uncertainty variables (u) and the available mitigation actions (m) that are available should the unmitigated design fail to meet one or more constraints.

Finally, the probability of success ($P\left(Success \mid x, h\right)$) and the expected value of the objective function ($\mathbb{E}[f(x, h)]$) are input into an optimizer described in Section 5.9.5. This optimizer will attempt to select a design (x, h) in order to minimize the expected value of the objective function such that the probability of success does not fall below a specified threshold ($Goal$). Until an optimum is found each new setting of design variables (x) and uncertainty margins (h) will be input into the Vehicle Sizing to iterate through the UQ&M algorithm. Upon finding an optimum, the design point (x, h) is recorded along with any relevant outputs from the algorithm.

5.9.1 Uncertainty Quantification Algorithm

This section provides more detail about the uncertainty quantification algorithm described in Section 5.9. In order to select the optimum design, the overall algorithm must contain component algorithms capable of calculating the expected value of the objective function as well as the probability of success ($P\left(Success \mid x, h\right)$). The algorithm used to size the vehicle to the design conditions (x, h) is described in Section 5.9.2. Later analyses evaluate the performance of this sized vehicle under different conditions. The reliability analysis in Section 5.9.3 takes the fixed sized vehicle and applies different uncertainty scenarios to it in order to calculate a probability of compliance $(P\left(Compliance \mid x, h\right))$ respect to the performance constraints as well as the expected value of the objective function $(\mathbb{E}[f(x, h)])$. The reliability analysis also collects the failed uncertainty scenarios (U_{Failed}) for later recovery tests. The mitigation analysis evaluates the individual failed uncertainty scenarios $(u \in U_{Failed})$ further and attempts to find a combination of mitigation actions (m) which will bring the design back into compliance with the performance constraints for that failed uncertainty scenario. The aggregate of these recovery attempts yields the probability of recovery of the design $(P\left(Recovery \mid x, h\right))$. The information is collected and given to an optimizer which will try to find the best combination of design variables (x) and uncertainty margins (h) to minimize the expected value of the objective function subject to a minimum level of probability of success $(P\left(Success \mid x, h\right))$.

5.9.2 Vehicle Sizing Algorithm

This section provides more detail about the vehicle sizing algorithm described in Section 5.9. To assess a new design point, the vehicle must first be sized. The vehicle sizing logic flow is shown in Figure 49. First a design point consisting of design variable settings (x) and uncertainty margins (h) is given to the sizing analysis from the optimizer. The range requirement for the design mission (R_{Des}) is also input into the analysis so the vehicle can be designed and sized for later analysis.

The engine is first designed based on the design variable input vector (x). This engine is given to the aircraft design algorithm. Once the initial parameters are set, a sizing mission is executed to evaluate the vehicle's performance. In this analysis the range requirement (R_{Des}) dictates the size of the vehicle, given the design point

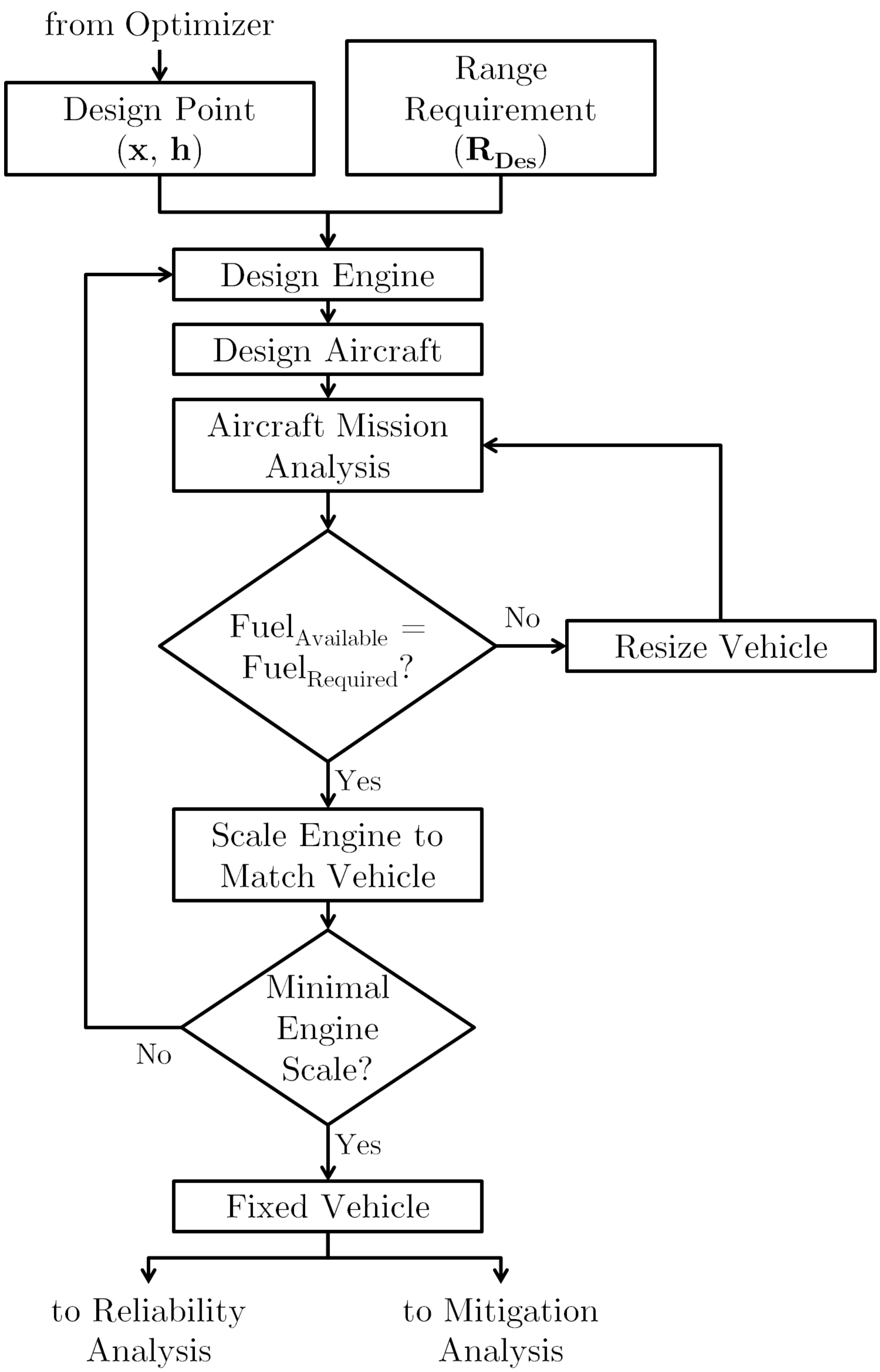


Figure 49: Vehicle Sizing Logic Flow

specified by the DoE. Once the mission has been evaluated, the resulting fuel required to complete the mission ($Fuel_{Required}$) is compared to the fuel capacity of the aircraft ($Fuel_{Available}$). If there is insufficient fuel available, the vehicle will be scaled up. Should there be too much fuel, the vehicle is downsized. This rescaled vehicle is reanalyzed by the aircraft mission analysis and the required and available fuel amounts are compared once again.

Once the fuel requirements match, the engine is scaled to meet the aircraft requirements. Should the engine need to be scaled by more than a small amount, the model will return to the engine design to create a new engine with the new required thrust targets. Once the engine thrust output from the mission analysis matches the thrust from the engine sizing, the vehicle is considered sized. The configuration is frozen, and the output fixed vehicle is given to the performance analysis.

5.9.3 Reliability Analysis Algorithm

This section provides more detail about the reliability analysis algorithm described in Section 5.9. After the vehicle has been sized, a reliability analysis is performed to determine the probability of compliance of the aircraft as designed. The logic flow of the reliability analysis is shown in Figure 50. The fixed vehicle which was output from the vehicle sizing algorithm is input into the analysis. The distributions of the uncertainty variables are brought into the analysis as a set of uncertainty scenarios (U). Performance constraint targets (g_{req}) are also input into the analysis in order to assess the aircraft's ability to meet these targets.

As mentioned in Section 5.9, a sample uncertainty scenario (u) is first drawn from the set of all uncertainty scenarios (U). The fixed aircraft's performance is evaluated under this uncertainty condition. If the aircraft's performance under this uncertainty scenario satisfies Equation (13) (i.e. it meets all constraints simultaneously), then the aircraft is considered compliant under that scenario (u).

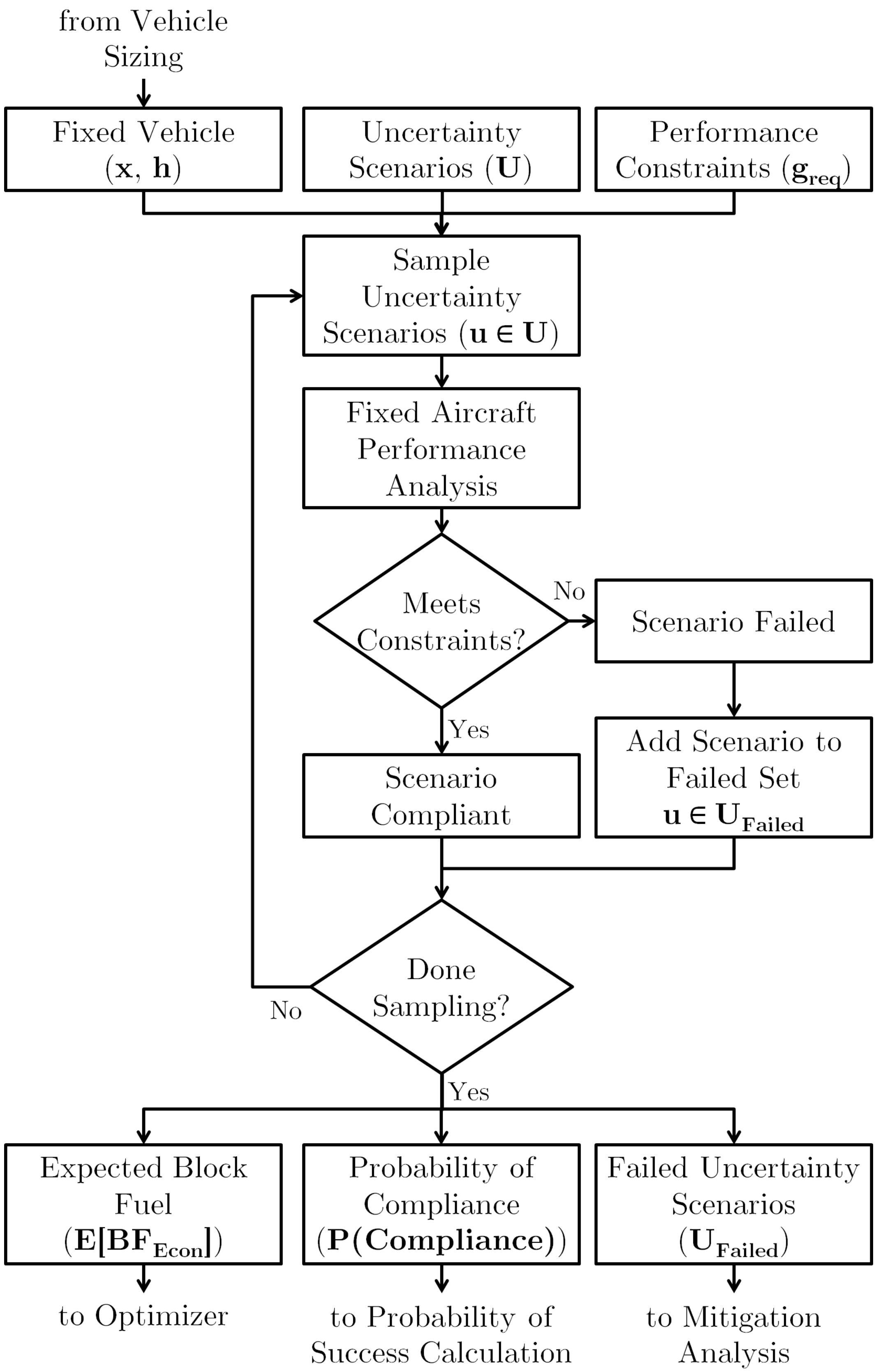


Figure 50: Reliability Analysis Logic Flow

$$(g_i(x,h,u,0) \geq 0)\,\forall i \in G \tag{13}$$

If, however, the aircraft performance does not meet the constraints, the vehicle is considered non-compliant under that uncertainty condition, and the uncertainty condition is added to the set of failed uncertainty scenarios ($u \in U_{Failed}$). After this determination has been made on the compliance of the vehicle under a particular set of uncertainty conditions, the algorithm checks to see if all uncertainty scenarios (U) have been exhausted. If sampling is not complete, a new sample scenario ($u \in U$) is drawn and the process is repeated.

Once all samples have been drawn, the expected value of the objective function ($\mathbb{E}[f(x,h)]$) is evaluated via Equation (25). This value will be given to the optimizer.

$$\mathbb{E}[f(x)] = \frac{\int_{u\in A} f(x,h,u,0)\,du + \int_{u\in A_M} f(x,h,u,m)\,du}{\int_{u\in A}(1)du + \int_{u\in A_M}(1)du} \tag{36}$$

The probability of compliance of the design ($P(Compliance \mid x,h)$) is calculated using Equation (17). The probability of compliance will be used in the probability of success ($P(Success \mid x,h)$) calculation later.

$$P(Compliance \mid x,h) = \frac{\int_{u\in U} 1_A(x,h,u)\,du}{\int_{u\in U} 1du} \tag{17}$$

Finally, the failed uncertainty scenarios (U_{Failed}) are passed to the mitigation analysis. These scenarios will be assessed using the available mitigation actions to determine whether or not the design's performance can be recovered under each scenario ($u \in U_{Failed}$).

5.9.4 Mitigation Analysis Algorithm

This section provides more detail about the mitigation analysis algorithm described in Section 5.9. Once the Reliability Analysis is complete, the Mitigation Analysis shown in Figure 51 assesses the probability of recovery ($P(Recovery \mid x,h)$) of the vehicle through a set of potential mitigation actions (M). The assessment takes as inputs a

Fixed Vehicle, a set of failed uncertainty scenarios, a set of mitigation actions, and the required performance constraints imposed on the aircraft. The Fixed Vehicle is an output from the Sizing Analysis which was based on only the design variables (x) and associated uncertainty margins (h). The set of failed uncertainty scenarios (U_{Failed}) comes from the Reliability Assessment and only includes uncertainty scenarios under which the proposed aircraft design would fail to meet one or more performance constraints. The performance constraints (g_{req}) are the same constraints imposed on the aircraft during the Reliability Assessment. An allowable set of mitigation actions (M) are input to allow the Mitigation Analysis to try to recover the aircraft for failed uncertainty scenarios.

After establishing all of the input values, a single scenario is drawn from the set of failed uncertainty scenarios (U_{Failed}). For this particular uncertainty scenario, one or more performance constraints was violated during the Reliability Assessment. A set of samples[2] are selected from the mitigation space to assess the possibility of recovering performance and bringing the aircraft back into compliance with all constraints, and a single sample is drawn from the mitigation space. For this sample, the performance of the Fixed Vehicle is assessed under the previously sampled failed uncertainty scenario. This assessment yields performance information which is compared to the performance constraints on the aircraft. If the aircraft under the failed uncertainty scenario now meets all constraints simultaneously once the sampled mitigation action is applied, the scenario is considered recovered. If, however, the aircraft still fails to meet one or more of the performance constraints, the uncertainty scenario is not yet recovered. Should remaining samples exist from the mitigation space, the algorithm will return to the mitigation sampling stage. Should no further mitigation action samples exist, it is assumed that the sampled uncertainty scenario is not recoverable through mitigation actions and is considered failed.

[2]Multiple alternative methods of sampling are discussed in Section 3.5.

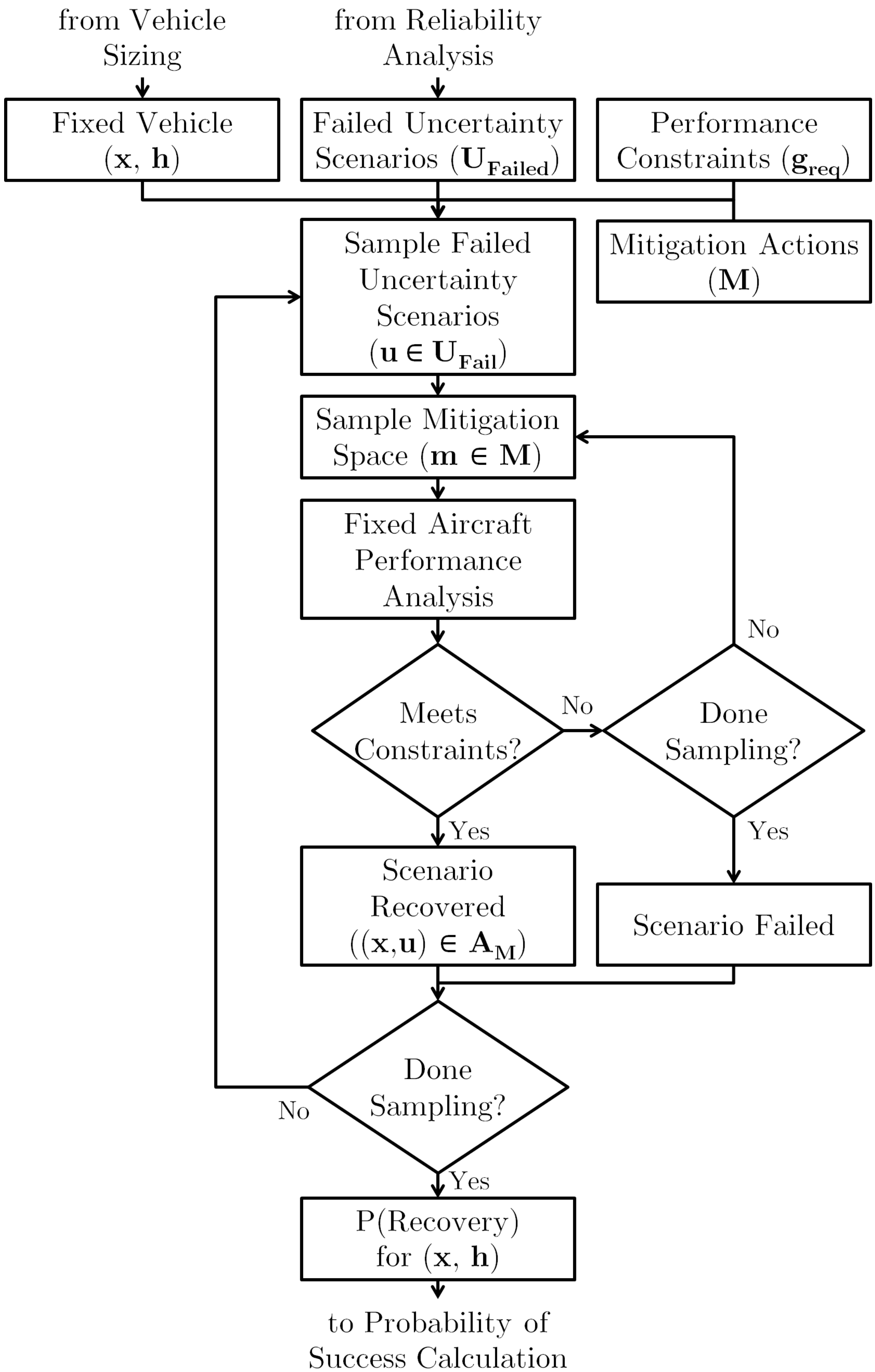


Figure 51: Mitigation Analysis Logic Flow

Once the recoverability of the particular sampled uncertainty scenario is established through either finding a set of mitigation actions that recover the design or exhausting the mitigation action samples, the algorithm checks to see if additional uncertainty scenarios exist. Should more failed uncertainty scenarios (U_{Failed}) exist, the algorithm will return to the loop that samples from these uncertainty scenarios to assess the recoverability of another uncertainty scenario. Once all failed uncertainty scenarios have been assessed, the probability of recovery ($P\left(Recovery \mid x,h\right)$) of the selected design is calculated. The Mitigation Assessment's probability of recovery is then provided to the optimization algorithm.

5.9.5 Final Calculations and Optimizer

This section provides more detail about the optimizer described in Section 5.9. The outputs of the Reliability Analysis and the Mitigation Analysis are brought together to calculate the final probabilistic performance of the vehicle. The overall probability of success ($P\left(Success \mid x,h\right)$) of the design is calculated via Equation (24) by combining the probability of compliance ($P\left(Compliance \mid x,h\right)$) from the reliability analysis with the probability of recovery ($P\left(Recovery \mid x,h\right)$) from the mitigation analysis. The resulting probability of success metric represents the likelihood that the given aircraft design (x and h) selected by the optimizer will be compliant with all performance constraints (g_{req}) given the distributions of the uncertainty variables (u) and the available mitigation actions (m) that can be used should the unmitigated design fail to meet one or more constraints.

$$\begin{aligned} P\left(Success \mid x,h\right) =& P\left(Compliance \mid x,h\right) \\ &+ P\left(Recovery \mid x,h\right)\left(1 - P\left(Compliance \mid x,h\right)\right) \end{aligned} \tag{24}$$

The probability of success ($P\left(Success \mid x,h\right)$) metric is then input into the optimizer. This metric will be used as a constraint within the global optimizer. Designs which fall below a specified threshold (indicated as *Goal* in Figure 48 will not comply with the optimization objective.

As stated in Section 5.9, this optimizer will attempt to select a design (x, h) in order to minimize the expected value of the objective function defined in Step 5. Again, the probability of success constraint will ensure that the probabilistic performance does not fall below the specified threshold. The optimizer will input new settings of design variables (x) and uncertainty margins (h) into the vehicle sizing algorithm to attempt to minimize the expected value of the objective function while maintaining a required level of success. Once the optimum is found, the design point (x, h) is recorded.

5.10 Step 10: Execute UQ&M Environment

The output of executing the Uncertainty Quantification and Management environment created in Step 9 is an optimal design based on the designer's preferences. Should only one specific set of goals be desired, this design should be sufficient. However, it is likely that a designer or company will not yet know exactly what goals to specify when setting up the problem. Thus, the UQ&M environment may need to be executed multiple times under different reliability goals or objective function weightings to yield a set of designs from which to select. If this is the case, many multi-attribute decision making (MADM) techniques exist within the existing literature to make such a selection.

5.11 Chapter Summary

In this chapter, a specific step-by-step formulation of the methodology proposed in Chapter 3 was developed. The details of the implementation were informed by the results of the hypothesis testing from Chapter 4. Details about how to decide whether the ARMOUR methodology is appropriate for the design problem under consideration were discussed. Directions were given on how to define aircraft requirements, uncertainty information, and mitigation actions. Suggestions were given on how to set appropriate objective metrics. Requirements on the designs space were given. Guidelines on modifying existing aircraft models were given along with directions on how to use these models to generate the necessary surrogate models. The uncertainty quantification and management environment central to the ARMOUR method was discussed in enough detail to allow an aircraft designer to implement the method independently. Finally, the text describes how to use this environment to select an appropriate design with which to move forward from conceptual design. In Chapter 6 this step-by-step version of the ARMOUR methodology will be implemented for a specific aircraft design problem.

CHAPTER VI

IMPLEMENTATION

This chapter describes the specific way in which the methodology developed in Chapter 5 was implemented. First, the overall needs of the methodology are described. These include the top-level optimizer, physics-based modeling, uncertainty analysis, input variables (design, uncertainty margins, uncertainty, mitigation), mitigation penalties, responses and constraints, and probability measures. The implementation in this chapter is designed to be more directly analogous to the kind of implementation ARMOUR would see in practical applications. It will make use of a monolithic aircraft design and analysis tool. Specific care will be taken to ensure that uncertainty and mitigation are implemented as required by the hypotheses and experiment results. Due to the multi-loop nature of this problem, surrogate models will be employed to vastly increase the speed of this otherwise cumbersome analysis. An optimizer can be used to explore the resulting trade space between the new metric, probability of success, and traditional design objectives.

6.1 Step 1: Establish the Need

Company Z has decided to manufacture a new aircraft. Specifically, they are looking at developing a large civil transport aircraft in the class of the A340-300 and the 777-200ER. They have determined that significant enough uncertainty exists within their conceptual level design tools that they are not confident enough to simply perform a deterministic design analysis. They are more confident in their preliminary design tools and expect that much of the uncertainty surrounding the design will be reduced long before the design moves to pre-production; while changes will not be desirable at that stage, they are possible if the need arises. However, the design will be frozen long before any of these preliminary design analyses take place. Furthermore, these preliminary design analyses take a long time to execute and require significant design specification before they can be performed, which prevents the conceptual design team from employing these analyses before the design freeze. Thus, Company Z decides that the Aircraft Recovery through Mitigation & Optimization under Uncertainty for Reliability (ARMOUR) method is an appropriate method to help them make decisions during conceptual design.

6.2 Step 2: Aircraft Requirements Definition

Company Z's market research indicates that a specific design mission is required for their new aircraft. The design mission selected is to transport three hundred and five (305) passengers a distance of 7,530 nautical miles. This mission should account for all Federal Aviation Administration (FAA) requirements. Additionally, it is desired that the vehicle meet other performance constraints to allow the aircraft to operate out of a large number of airports.

In order to measure the reliability and recoverability of a design, performance constraints need to be established. These performance constraints will be measured with respect to a response from the aircraft model, so appropriate responses must be selected. These responses should be indicative of performance constraints which would be relevant to conceptual aircraft design. Furthermore, at least some of the responses need to be dependent on the uncertainty variables modeled in Section 6.3. If none of the responses of interest are affected by the uncertainty variables, then the reliability assessment will be unnecessary; a deterministic performance assessment would achieve the same answer.

To track fuel consumption, the block fuel required for an economic range mission (BF_{Econ}) will be considered. This metric will represent the fuel efficiency of the resulting aircraft. Economic range block fuel will serve as a proxy for the operating cost of the aircraft - an important metric to consider when selling an aircraft. It is desired to minimize this response. As such, no specific goal will be established for the economic range block fuel.

The aircraft will be sized to meet the specified range ($RANGE$) of the design mission. However, once an uncertainty scenario is considered, the vehicle may no longer be capable of meeting the range requirement when carrying the design payload. Thus, the range of the vehicle at the design payload must be enforced. This constraint will be the same as the sizing requirement, so range for the design mission shall not be less than 7,530 nautical miles.

The approach speed (V_{App}) of the aircraft will be important from an airport access perspective. This will dictate at which airports the aircraft can land, potentially restricting the desired city pairs for prospective buyers. As such, a maximum approach speed should be specified by the designer. For this problem, a constraint shall be imposed on the aircraft such that the approach speed may not exceed 145 knots.

Takeoff field length ($TOFL$) also dictates whether an aircraft can operate from a given airport. Obviously, the takeoff field length must be less than the available field length present at any desired departure airports. For this study the takeoff field length of the vehicle shall not exceed 11,000 feet.

Wing span ($Span$) will also dictate the airports at which the vehicle can operate. The span will limit access to gates and may even prevent the aircraft from using the runway at all. This limit will counter the aerodynamically desirable tendency to increase the aspect ratio. For this study the wing span shall not be allowed to exceed 215 feet.

The rate of climb, a.k.a. the excess power, dictates the aircraft's ability to change altitude. There must be sufficient available climb rate at the design altitude to enable the aircraft to maneuver in order for it to operate in this region effectively. A constraint is imposed such that the climb rate shall not be less than 300 feet per minute at an altitude of 35,000 feet.

The ARMOUR method, like all reliability-based methods, specifies a required minimum level of probability to meet constraints. For this problem, probability of compliance ($P(Compliance \mid x, h)$) and probability of success ($P(Success \mid x, h)$) with respect to all constraints listed in Table 2 will simultaneously be imposed. The levels of reliability desired will be specified in Step 5.

Table 2: Performance Constraints and Traditional Objective

Metric	Symbol		Target	Units
Approach Speed	V_{App}	$\leq$	145	knots
Takeoff Field Length	$TOFL$	$\leq$	11,000	feet
Rate of Climb at 35,000 ft	RoC	$\geq$	300	feet per minute
Design Mission Range	$Range$	$\geq$	7,530	nautical miles
Wing Span	$Span$	$\leq$	215	feet
Economic Mission Fuel (4000 nm)	BF_{Econ}		minimize	pounds

These metrics are standard aircraft performance measures. Approach speed is a common measure of the landing capability of an aircraft and is regulated by the FAA. As mentioned, takeoff field length will dictate the airports where the aircraft can operate and the rate of climb is another regulation requirement. The range for the design payload is a prime indicator of the markets for which the aircraft can be sold. Economic mission fuel is not constrained as this will be the "traditional design metric" used in the Pareto Frontier, along with the probability of success.

6.3 Step 3: Elicit Uncertainty Information

Uncertainty variables are necessary to emulate the true uncertainty inherent in the design process. These variables are implemented only after design is complete; they represent the aircraft drifting from its assumed performance condition during later design stages. Emulating a specific type of uncertainty can be accomplished in different ways. For example, the component weight uncertainty can be modeled by implementing a distribution about the weight of each component individually (e.g. wing weight, fuselage weight, empennage weight, landing gear weight, engine weight, etc.). These component weights may each have their own associated uncertainty distribution, owing to the individual level of detail contained in their individual calculation methods during conceptual design. These component weights may also be correlated with each other if there exists a logical or historical reason that the error in one prediction typically correlates with another error. Alternately, an assumed distribution on the overall empty weight of the vehicle can be implemented which accounts for the distribution of all individual errors and their correlations.

For the implementation within this body of work, only a handful of uncertainty variables were selected. It is desired that these variables represent a large enough impact on the performance of the vehicle to test this method. Thus, factors important to the overall performance of the aircraft like the empty weight, cruise drag, engine fuel flow, and Mach number at drag divergence were considered. As discussed, these factors may be resolved into individual component uncertainties with separate and potentially correlated distributions, or they may be characterized by a gross overall distribution which emulates the combined effect of multiple components simultaneously. Table 3 contains the uncertainty variables used in these analyses.

Table 3: Uncertainty Variables

Variable Description	Symbol	Minimum	Maximum
Empty Weight Error	u_{EW}	-1%	6%
Cruise Drag Error	u_{Drag}	-1%	6%
Fuel Flow Error	$u_{FuelFlow}$	-1%	6%
Mach Drag Divergence Error	u_{MDD}	-5%	5%

The generation of a sampled set of uncertainty scenarios (U) can be performed outside of the uncertainty quantification algorithm described in Section 6.9. Using Monte Carlo simulation (MCS) as a sampling method, an assumed distribution may be modeled by generating an appropriate set of samples; as the full set of samples grows larger, the desired distribution is approximated as discussed in Section 2.4.1.

These samples are created using a pseudo random number generator available in almost any modern computer code. Since pseudo random number generators by default create independent uniform distributions ranging from zero to one, a transformation is necessary to modify the generated distributions into the desired ones. If correlation between the distributions is desired, a Cholesky decomposition may be used to modify the distributions appropriately.

Without an existing team of expert engineers or a set of separate conceptual and preliminary design tools to compare, assumptions need to be made about the distributions of uncertainty variables. For this work, all uncertainty variables were given uniform distributions between their respective minimum and maximum values. This was implemented because some uncertainty distribution must be assumed to perform the reliability analyses. If uncertainty quantification analysis were performed to yield a better assumed distribution form, then those distributions could be employed using appropriate transformations. Since the assumed random variables in this implementation are all independent and uniformly distributed, only a simple scaling of the computer-generated uniform distributions ($u_{j_{rng}}$) in Equation (37) is necessary to create distributions which match the desired interval (u_j). Once the set of scenarios is established, it will be used by the fixed aircraft analysis as-is and need not be regenerated.

$$u_j = u_{j_{rng}} * \left(u_{j_{UpperLimit}} - u_{j_{LowerLimit}}\right) + u_{j_{LowerLimit}} \tag{37}$$

It is important to consider how the uncertainty variables will affect the responses. From Equation (34) it is apparent that final approach speed of the aircraft before landing (V_{App}) is dependent on the landing weight of the aircraft, the vehicle's wing area, the maximum lift coefficient of the aircraft, and the local air density. The local air density is assumed to be a constant value based on the sea-level conditions of the ICAO Standard Atmosphere and will not vary in this implementation. Wing area is set by the sizing analysis and is expected to be frozen during later design stages, where uncertain values will be realized. The maximum lift coefficient may be modified by adjusting the high-lift devices installed on the vehicle, but these devices are not explicitly affected by any uncertainty variables. The landing weight of the vehicle is primarily composed of the payload weight, crew weight, remaining fuel at landing, and the operating empty weight of the vehicle. The payload and crew are fixed weights for a commercial transport mission. The effect of any selected uncertainty variable on the remaining fuel in the vehicle will be small. However, the operating empty weight of the vehicle is directly affected by the empty weight uncertainty (u_{EW}) selected in Section 6.3. Thus, as represented by Equation (38), approach speed (V_{App}) is a function of the design variables (x), uncertainty margins (h), and the empty weight uncertainty.

$$V_{App} = f\left(x, h, u_{EW}\right) \tag{38}$$

The takeoff field length (TOFL) of a vehicle is primarily determined by the takeoff gross weight (TOGW), engine thrust, and the low speed aerodynamics of the vehicle. Currently, the takeoff gross weight for all evaluation conditions is set by the maximum takeoff weight (MTOW) established by the design mission during sizing. Under this construct, TOGW will be a function of only design variables (x) and margins (h); uncertainty variables (u) will have no impact on the TOGW of the vehicle. The thrust performance of the engine is determined during design. All degradations possible due to uncertainty in the current setup of the analysis impact only the fuel flow characteristics of the engine. Thus, thrust is not affected by uncertainty. The low speed aerodynamics also are not affected by uncertainty in the current problem. Thus, TOFL is only a function of the sizing conditions of the aircraft, namely design variables and margins, as shown in Equation (39).

$$TOFL = f\left(x, h\right) \tag{39}$$

Equation (40) shows the expected functional relationship between the rate of climb

(RoC) and the input variables. The drag uncertainty variable (u_{Drag}) is expected to impact the rate of climb both directly and through the change in fuel consumed to arrive at the top of climb condition. The Mach drag divergence variable (u_{MDD}) will have a similar impact to the drag changes. The fuel flow uncertainty variable ($u_{FuelFlow}$) will change the rate of climb only through the impact to the fuel consumed to arrive at top of climb.

$$RoC = f\left(x, h, u_{Drag}, u_{MDD}, u_{FuelFlow}\right) \tag{40}$$

The vehicle wing span ($Span$) is not expected to be impacted by any of the uncertainty variables. Thus, the wing span will only be a function of the design variables (x) and the uncertainty margins (h), as shown in Equation (41). This means that a constraint imposed upon wing span will be deterministic.

$$Span = f\left(x, h\right) \tag{41}$$

The range of the vehicle for a design mission ($RANGE$) is expected to be impacted by all of the uncertainty variables. Empty weight uncertainty (u_{EW}) will increase or decrease the fuel available to fly the mission. Drag uncertainty (u_{Drag}) will increase or decrease the thrust required to maintain speed, which will change the fuel required to maintain speed. The uncertainty on Mach drag divergence (u_{MDD}) will also affect the thrust requirements. The fuel flow uncertainty ($u_{FuelFlow}$) will change the fuel consumed per unit of thrust required; this will change the fuel economy of the vehicle, reducing the range capability. Equation (42) shows the functional relationship between design mission range ($RANGE$) and these variables.

$$Range = f\left(x, h, u_{EW}, u_{Drag}, u_{MDD}, u_{FuelFlow}\right) \tag{42}$$

The block fuel for an economic range mission (BF_{Econ}) will be impacted by all uncertainty variables. Empty weight uncertainty (u_{EW}) will change the weight of the aircraft during the mission. Drag uncertainty (u_{Drag}) will increase or decrease the thrust required to maintain speed; this will change the fuel required to maintain speed, changing the fuel consumption. The uncertainty on Mach drag divergence (u_{MDD}) will also affect the thrust requirements. The fuel flow uncertainty ($u_{FuelFlow}$) will change the fuel consumed per unit of thrust required, changing the fuel economy of the vehicle. Equation (43) shows the functional relationship between the block fuel for an economic range mission (BF_{Econ}) and these variables.

$$BF_{Econ} = f\left(x, h, u_{EW}, u_{Drag}, u_{MDD}, u_{FuelFlow}\right) \tag{43}$$

6.4 Step 4: Define Potential Mitigation Actions

For this thesis, there is no direct access to an aircraft manufacturer's design philosophies, so basic aerospace engineering concepts will be used to construct a list of mitigation actions. The goal is to show how mitigation actions would be implemented, rather than to provide guidance on the specific mitigation actions that should be used. To accomplish this, it is not necessary to figure out exactly which mitigation actions would be used by any specific company; rather, as long as representative mitigation actions have been selected with the rules set forth in Section 3.3.2 in mind, they will be able to demonstrate the process. Were an aircraft company to implement the ideas of this thesis it would be more appropriate to employ their own design philosophy and engineers to inform them which mitigation actions to investigate and how best to model their benefits and possible penalties.

Mitigation actions are established under the assumption that they will be non-ideal, late stage changes to a design. Sections 2.2.2 and 3.3.1 have discussed the reasoning and need for penalties on possible mitigation actions to accurately portray the negative side-effects of these actions.

6.4.1 Post-Sizing Fuel Addition

The first mitigation action modeled in this thesis is a post-sizing fuel addition in which a "topping off" of the fuel tanks is considered to fix range constraint violations. Fuel will be added to the vehicle after the sizing loop is completed. This additional fuel will cause the aircraft's maximum takeoff weight (MTOW) to increase beyond the design condition, adversely affecting the performance of the vehicle. This extra fuel will, of course, allow the aircraft to fly further than it could at the design MTOW. The fuel capacity will be increased by up to at most ten percent of the baseline vehicle's mission fuel weight – 28,000 lbs. The extra MTOW is expected to have an adverse effect on any takeoff and climb constraints. This behavior should be captured by any physics-based model.

This fuel increase will of course be limited by the capacity of the fuel tanks. In other words the aircraft may not be able to be mitigated up to the full amount, depending on the design of the vehicle – specifically the size of the fuel tanks. This is expected to have an effect on the probability of recovery, since the fuel tank size will dictate the maximum effectiveness of this mitigation action. This limit will be imposed by allowing the aircraft performance assessment tool to calculate the available fuel tank size of the vehicle, which scales with the wing geometry.

The increase in the maximum takeoff weight of the vehicle associated with adding fuel will have adverse effects on the structure of the vehicle. Since the aircraft was only designed to handle at most the design MTOW, it is expected that this increase in load will force the structural weight of the vehicle to increase. This structural weight increase will be enforced via an Operating Empty Weight (OEW) penalty. This empty weight penalty ($EW_{penalty}$) will be proportional to the change in maximum takeoff weight requested for mitigation as shown in Equation (44). This new MTOW will include both the additional fuel and the corresponding empty weight addition. Equation (44)

$$m_{MTOW} = MTOW_{new} - MTOW_{DES} \tag{44}$$

A relationship was assumed between this additional maximum takeoff weight during preliminary design and the empty weight penalty. This penalty function should behave in such a way that minimal penalty would be incurred for small changes to the aircraft. After this initial burn-in period, the penalty should monotonically increase as the aircraft deviates from its design condition. Furthermore, the rate of penalty is expected to increase as the aircraft grows excessively. Thus, a quadratic equation was settled upon, as shown in Equation (45).

$$EW_{penalty}\left(m_{MTOW}\right) = maximum\left(0, 0.0002608 * m_{MTOW}^2 - 0.0063233 * m_{MTOW}\right) \tag{45}$$

The empty weight penalty function in Equation (45) is plotted in blue against the change in maximum takeoff weight in Figure 52. This blue line shows the empty weight increasing at a greater and greater rate as the aircraft deviates from its design weight. It is also possible to determine the actual fuel associated with an increase in $MTOW_{MA}$. To do so the empty weight penalty can simply be subtracted from the change in maximum takeoff weight ($MTOW_{MA} - EW_{penalty}$), yielding the remainder as the change in fuel weight. By doing so, it is possible to determine the empty weight penalty per pound of fuel added to the aircraft. This fuel weight increase is shown in red.

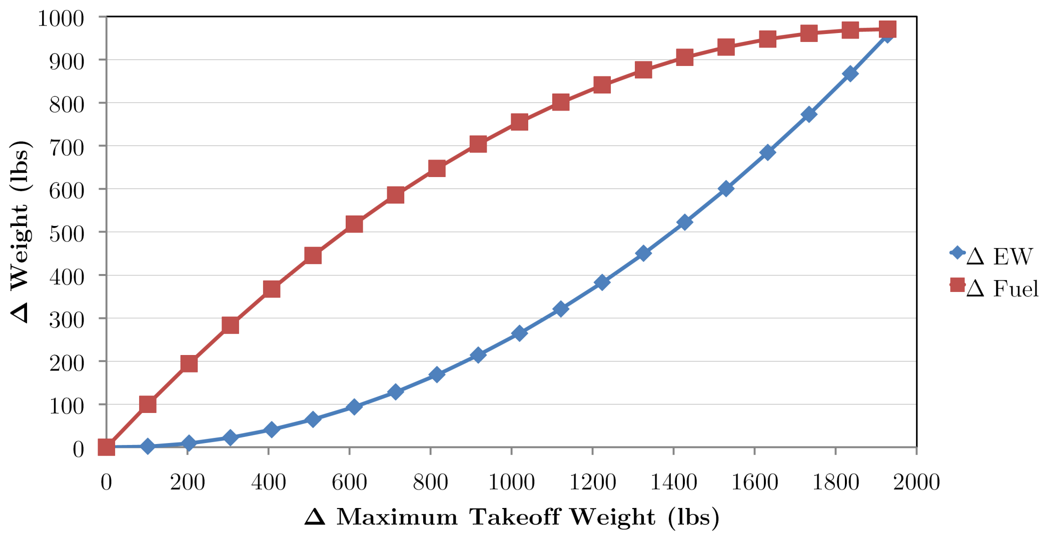


Figure 52: Empty Weight Penalty per Pound of Maximum Takeoff Weight

Because the empty weight penalty increases in intensity as MTOW grows larger, the effectiveness of the mitigation action (the amount of fuel being added) decreases. This relationship is shown in Figure 53. Here, the empty weight penalty $EW_{penalty}$ from Equation (45) is plotted on the y-axis. The x-axis is the actual fuel added to the aircraft – the difference between the additional maximum takeoff weight and the empty weight penalty. It is clear from this chart that the penalty imposed makes adding more than about 900 pounds of fuel to the aircraft increasingly ineffective.

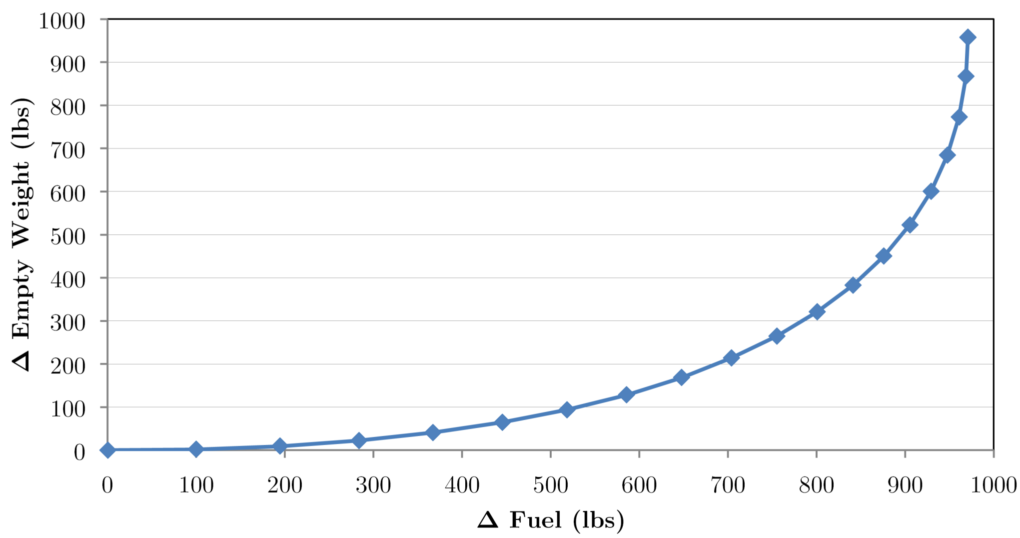


Figure 53: Derived Empty Weight Penalty per Pound of Fuel

6.4.2 Augmented High-Lift Device

The second mitigation action modeled in this thesis is to add a representative advanced high-lift device to the aircraft to counter approach speed constraint violations. This high lift device is expected to increase the maximum lift coefficient of the vehicle during landing. Based on Equation (34) from Section 4.2.1, this increase in the $C_{L_{max}}$ of the vehicle during landing will reduce the required approach speed.

This increase in lift will not come without a cost. The additional lift on the wing constitutes an additional load during landing. The wing, the landing gear, and other structural components may need to be reinforced to accommodate the impact of this mitigation action. Thus, the additional lift coefficient is expected to incur an empty weight penalty. Again, the penalty will be based on the change in lift coefficient from the design value as in Equation (46).

$$m_{C_L} = CL_{max_{landing_{new}}} - CL_{max_{landing}} \tag{46}$$

A penalty was assumed for this study as shown in Equation (47).

$$EW_{penalty}\left(m_{C_L}\right) = \frac{85000}{7} * C_{L_{MA}}^2 + \frac{3250}{7} * C_{L_{MA}} - \frac{3}{7} \tag{47}$$

The impact of Equation (46) on the aircraft is shown via Figure 54. In this diagram the final landing lift coefficient is shown on the x-axis versus the change in empty weight on the y-axis.

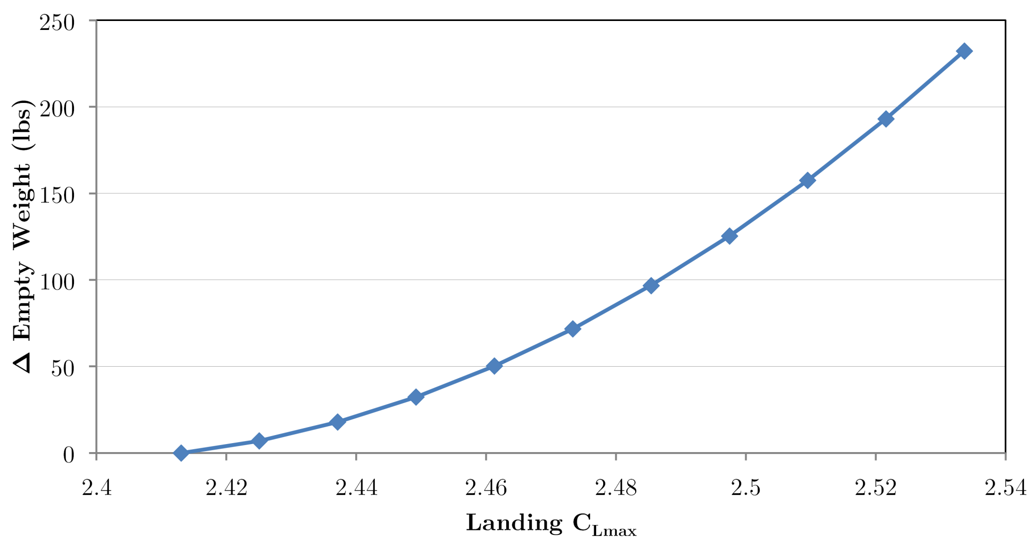


Figure 54: Landing Lift Augmentation Penalty

6.4.3 Engine Throttle Push

The final mitigation action modeled in this thesis is an engine throttle push to counter potential Takeoff Field Length ($TOFL$) and Rate of Climb (RoC) violations. An engine throttle push is accomplished by uprating the engine thrust without changing the size or general design of the engine, making it a good candidate for a mitigation action. This is accomplished by changing the engine controls such that the engine is allowed to run at a hotter temperature. This higher temperature is the result of adding additional fuel to the engine at maximum thrust. This increase in temperature will have a detrimental impact on the engine life span which is not modeled in this work.

As justification for the range of throttle push allowed for mitigation, engines in service were examined. The Pratt & Whitney 4098 engine is a throttle push variant of the PW4000 series [67]. This engine is of the same class as engines currently in use on the 777 series of vehicles [41]. To determine the intensity of the throttle push employed, the thrusts of the engine were compared to others in its family. The PW4098 is capable of 99,040 pounds of thrust when the stationary engine is measured at full power and sea-level conditions [22]. Since the next highest thrust of 91,790 pounds was achieved with the PW4090 variant, it is assumed that a throttle push of roughly eight percent was achieved. Considering the high level of throttle push seen with this engine, which is of the same class as those on the theoretical aircraft being implemented, a throttle push of five percent seems reasonable.

Because the takeoff field length will not be affected by the uncertain parameters, the constraint on TOFL is effectively deterministic. All designs (x and h) will either be one hundred percent or zero percent in compliance with the TOFL constraint. Designs which are zero percent compliant will be undesirable according to either a real design team or to any optimizer taking the probability of compliance ($P\left(Compliance \mid x, h\right)$) into account. Thus, the throttle push mitigation action is of little use in its normally expected role.

The throttle push mitigation action will still recover scenarios where the vehicle fails to meet rate of climb (RoC). It can also combine with the fuel capacity mitigation action to maintain compliance with TOFL under scenarios in which the aircraft needs additional fuel to meet range.

6.4.4 Mitigation Action Summary

Table 4 shows a summary of the mitigation actions to be used in the remainder of this work. Each of the three mitigation actions discussed above are shown. The constraint each mitigation action addresses, its primary impact on the aircraft model, and the applicable limit are also included.

Table 4: Mitigation Actions

Mitigation Action	Metric Addressed	Parameter	Limit
Fuel Increase	Design Mission Range	MTOW	28,000 lbs
Advanced High Lift Device	Approach Speed	$C_{L_{max}}$	0.1
Throttle Push	Takeoff Field Length	Thrust	5%

It should be noted that the maximum takeoff weight is assessed significantly past the boundary suggested by Figure 52. This boundary is based on the assumed Equation (45). In Section 7.5 this assumed equation will be changed, dictating a new boundary. This will necessitate the additional variability needed in the MTOW variable.

The functional relationship between approach speed (V_{App}) and input variables from Equation (38) is reformulated to account for mitigation actions in Equation (48). The augmented high lift device mitigation action is designed to directly improve the approach speed. Additionally, the logic behind the mpty weight variation impacting the approach speed holds true whether in regards to uncertainty (i.e. u_{EW}) or due to a penalty on the mitigation actions ($EW_{penalty}$).

$$V_{App} = f\left(x, h, u_{EW}, m_{C_L}, EW_{penalty}\right) \tag{48}$$

The functional relationship between the takeoff field length ($TOFL$) and the input variables from Equation (39) is reformulated to account for mitigation actions in Equation (49). The additional maximum takeoff weight from the added fuel mitigation action (m_{MTOW}) will degrade the resulting takeoff field length. This means that the mitigation action could cause constraint violations when used to recover range.

The throttle push mitigation action ($m_{Throttle}$) will have a direct beneficial effect on the takeoff field length and may be used to recover scenarios when additional fuel is required but the takeoff field length is close to its performance constraint.

$$TOFL = f\left(x, h, m_{MTOW}, m_{Throttle}\right) \tag{49}$$

The functional relationship between the rate of climb (RoC) and the input variables from Equation (40) is reformulated to account for mitigation actions in Equation (50). The throttle push mitigation action will have a direct impact on the thrust available at the top of climb condition, impacting the excess power available and, therefore, the rate of climb achievable by the vehicle.

$$RoC = f\left(x, h, u_{Drag}, u_{MDD}, u_{FuelFlow}, m_{Throttle}\right) \tag{50}$$

The range of the vehicle for a design mission ($RANGE$) from Equation (42) is updated in Equation (51) to include the impacts of mitigation variables. The addition of post-sizing fuel (m_{MTOW}) is intended to increase the vehicle's range simply by adding more fuel. Any empty weight penalty ($EW_{Penalty}$) on the aircraft will degrade the vehicle's range by requiring the aircraft to carry more weight throughout the mission, leaving less usable fuel weight.

$$Range = f\left(x, h, u_{EW}, u_{Drag}, u_{MDD}, u_{FuelFlow}, m_{MTOW}, EW_{Penalty}\right) \tag{51}$$

None of the mitigation actions themselves are expected to impact economic range block fuel (BF_{Econ}) significantly. However, the empty weight penalty ($EW_{Penalty}$) imposed by these mitigation actions will cause the aircraft to carry more weight throughout the mission, increasing the lift required to maintain altitude. The increase in lift will increase the drag on the vehicle, increasing the required thrust. This increase in thrust will require more fuel to be burned. This will alter the functional relationship of economic range block fuel seen in Equation (43) to that seen in

Equation (52).

$$BF_{Econ} = f\left(x, h, u_{EW}, u_{Drag}, u_{MDD}, u_{FuelFlow}, EW_{penalty}\right) \tag{52}$$

6.5 Step 5: Select Performance and Reliability Objectives

For this implementation, only performance objectives are used. However, since the cost to operate the aircraft is a significant factor in the marketability of the vehicle, a proxy for operating cost will be used - specifically, the block fuel needed for an "economic range" mission of 4,000 nautical miles.

In an effort to explore different scenarios in this work, multiple settings will be used for both the probability of compliance ($P\left(Compliance \mid x, h\right)$) and probability of success ($\succ$) to generate different results. In general, the probability of compliance requirement will range between greater than or equal to 75 percent and up to 100 percent. An enforced probably of compliance requirement will never be less than 70 percent in this study. The probability of success requirement will also vary between greater than or equal to 75 percent and up to 100 percent for studies within this work. The probability of success requirement will never be set below the probability of compliance requirement, as this would be meaningless. No reliability requirements for individual constraints have been imposed for this study.

6.6 Step 6: Establish Design Space

To emulate a reasonable design process, variables which affect different aspects of the resulting aircraft are needed. In general, it would be beneficial to have some gross aircraft-level design variables, some engine-related design variables, some wing-related design variables, and some uncertainty margins. Table 5 lists the conceptual design variables which have been selected for this study.

The implementation of many of these variables is straightforward. Most aircraft design tools already have the ability to directly input variables like thrust-to-weight,

Table 5: Conceptual Design Variables and Impacts

Variable Description	Symbol	Related Aspect(s)
Design SLS Thrust-to-Weight	T/W	Design, Engine
Design Wing Loading	W/S	Design, Wing
Wing Aspect Ratio	AR	Design, Wing
Engine Lapse Rate	LapseRate	Design, Engine
Engine Fan Pressure Ratio	FPR	Engine
Engine Overall Pressure Ratio	OPR	Engine
Empty Weight Margin	h_{EW}	Margin
Drag Margin	h_{Drag}	Margin, Wing
Fuel Flow Margin	$h_{FuelFlow}$	Margin, Engine

wing loading, and aspect ratio into either a sizing or performance analysis. An engine design tool should have the ability to receive thrusts, pressure ratios and lapse rates either directly or through trivial variable transformations. By linking the two codes together in a sizing loop, the engine and aircraft can be matched.

The uncertainty margins need to be implemented directly into the aircraft sizing code. However, care must be taken with these variables. Depending on the analysis mode, the same variables can behave as either uncertainty margins or as uncertainty variables, as theorized in Section 3.1. This concept will be further tested in Section 4.1. Based on the discussion from Hypothesis 2, these margins must be applied to the aircraft during the sizing analysis, and their values removed before the performance analysis. To successfully implement the uncertainty margins, the aircraft must be sized with these margins in place. Thus, they will need to be set and input into the aircraft analysis code during sizing. However, these variables must not impact the uncertainty variables directly. This concept is further explained in Section 6.7.2.

Ranges were derived relative to the baseline aircraft. Wing loading and lapse rate were varied by roughly plus or minus five percent of the baseline value. All other non-margin variables were given a variation of about ten percent of the baseline value. All ranges were rounded to only a few significant figures. The uncertainty margin variables were allowed to vary between no margin (0%) up to six-percent

(6%). Table 6 contains all design variables, their baseline values, and their associated ranges.

Table 6: Conceptual Design Variables and Ranges

Variable Description	Symbol	Baseline	Minimum	Maximum
Design SLS Thrust-to-Weight	T/W	0.296	0.26	0.34
Design Wing Loading	W/S	133.34	126.0	140.0
Wing Aspect Ratio	AR	10	9.0	11.0
Engine Lapse Rate	LapseRate	0.2014	0.18	0.22
Engine Fan Pressure Ratio	FPR	1.50	1.425	1.575
Engine Overall Pressure Ratio	OPR	42.58	38.32	46.84
Empty Weight Margin	h_{EW}	3%	0%	6%
Drag Margin	h_{Drag}	3%	0%	6%
Fuel Flow Margin	$h_{FuelFlow}$	3%	0%	6%

6.7 Step 7: Create Aircraft Model

A tool or set of tools to model conceptual aircraft design must be selected in order to demonstrate the ARMOUR methodology. Ideally, this tool should represent the kind of capabilities that are present in industry-style tools to show ease of implementation of this methodology as well as any potential challenges.

The ARMOUR method for aircraft conceptual design begins with a modeling and simulation (M&S) environment for vehicle sizing and performance analysis. It is assumed that aircraft companies will have their own set of legacy aircraft analysis tools which they will use during conceptual design. Since these tools will vary between companies and are unavailable for the current research, a tool with similar constraints will instead be employed. The Flight Optimization System (FLOPS) is a deterministic computer program created at the NASA Langley Research Center designed for conceptual and preliminary design of aircraft. [59]

FLOPS can be executed in sizing mode to predict the aircraft size and fuel burn for a prescribed design mission. In analysis mode, FLOPS will take in a fixed aircraft and predict response metrics such as approach speed (V_{App}), takeoff field length ($TOFL$), and range at design payload (*Range*). These modes are consistent with the modeling tools described in Section 2.1.3. As such, they can be arranged and utilized in either way described in Section 3.1.3, and thus will be sufficient for the needs of this thesis.

An engine model was constructed in NASA's Numerical Propulsion System Simulation (NPSS) tool [66]. This tool was selected because it can both design and evaluate the performance of a turbofan engine. An engine model was first calibrated to a GE90-94B engine model [22]. This engine is commonly used on the Boeing 777 [41]. The engine model was altered to show improvements representative of modern engine advancements (i.e. slightly higher overall pressure ratio, slightly increased component efficiencies) to generate the new engine baseline. NPSS only calculates engine thermodynamics; thus, to calculate the physical dimensions and weight of the engine, another analysis tool was required. Weight Analysis for Turbine Engines (WATE) is one such engine turbomachinery calculation code [88]. It is heavily integrated with NPSS, allowing the outputs from NPSS to be directly read as inputs. WATE is used to calculate the weights and dimensions of the engine. The resulting engine performance, dimensions, and weight can be input into an aircraft sizing and analysis tool like FLOPS.

6.7.1 Baseline Aircraft Model

The selection of a baseline vehicle will help to establish appropriate input variable ranges, responses, and performance constraints for evaluation. For this implementation, a civil transport aircraft has been selected. The implementation herein will model a 305 passenger civil transport similar to a Boeing 777-200ER [10, 41]. The general behavior of this aircraft is well-established in the literature. This class of vehicle is considered relevant because of its prevalence in the national airspace. Furthermore, future environmental regulations and constant economic pressure will likely cause a redesign of this vehicle in the future.

However, this thesis is motivated by the concept of designing a new vehicle and the associated uncertainty. Calibration to a known vehicle is insufficient because there is little uncertainty associated with its performance. Instead, the baseline vehicle concept is modified to emulate a new design baseline. This new design point will be used as the baseline for all data generation from the physics-based model. Table 7 shows the baseline vehicle information which is used to calibrate the models compared to a 777-200ER. The calibrated model is used as the reference or baseline point for generating surrogate models.

Table 7: 300 Passenger Aircraft Baseline Compared to 777-200ER Model[10]

Metric	Baseline	777-200ER	Units
Capacity	301	301	Passengers
Maximum Take-Off Weight (MTOW)	561,697	656,000	pounds
Operating Weight Empty (OWE)	276,875	302,200	pounds
Maximum Zero Fuel Weight	340,085	430,000	pounds
Maximum Fuel Capacity	33,608	45,200	gallons (U.S.)
Maximum Fuel Capacity	224,754	302,270	pounds
Maximum Structural Payload	125,550	125,550	pounds
Design Mission Range	7,530	7,530	nautical miles
Payload (Passengers + Baggage)	63,210	63,210	pounds
Wing Span	196.3	199.9	feet
Sea Level Static Thrust per Engine	83,300	93,700	pounds of force
Cruise Mach	0.84	0.84	N/A

6.7.2 Detailed Modeling Changes

In order to determine the aircraft performance, the vehicle must first be sized. The model sizing analysis is shown in Figure 55. First the model must take in a design point from the Design of Experiments. This design point will contain a specific setting of design variables (x) and uncertainty margins (h). Along with the range requirement (R_{Des}), this information is all that will be needed to size the vehicle so that a fixed vehicle can be input into later analyses.

The engine thermodynamic sizing step takes in this design point and designs the engine using the Numerical Propulsion System Simulation (NPSS) code. Once

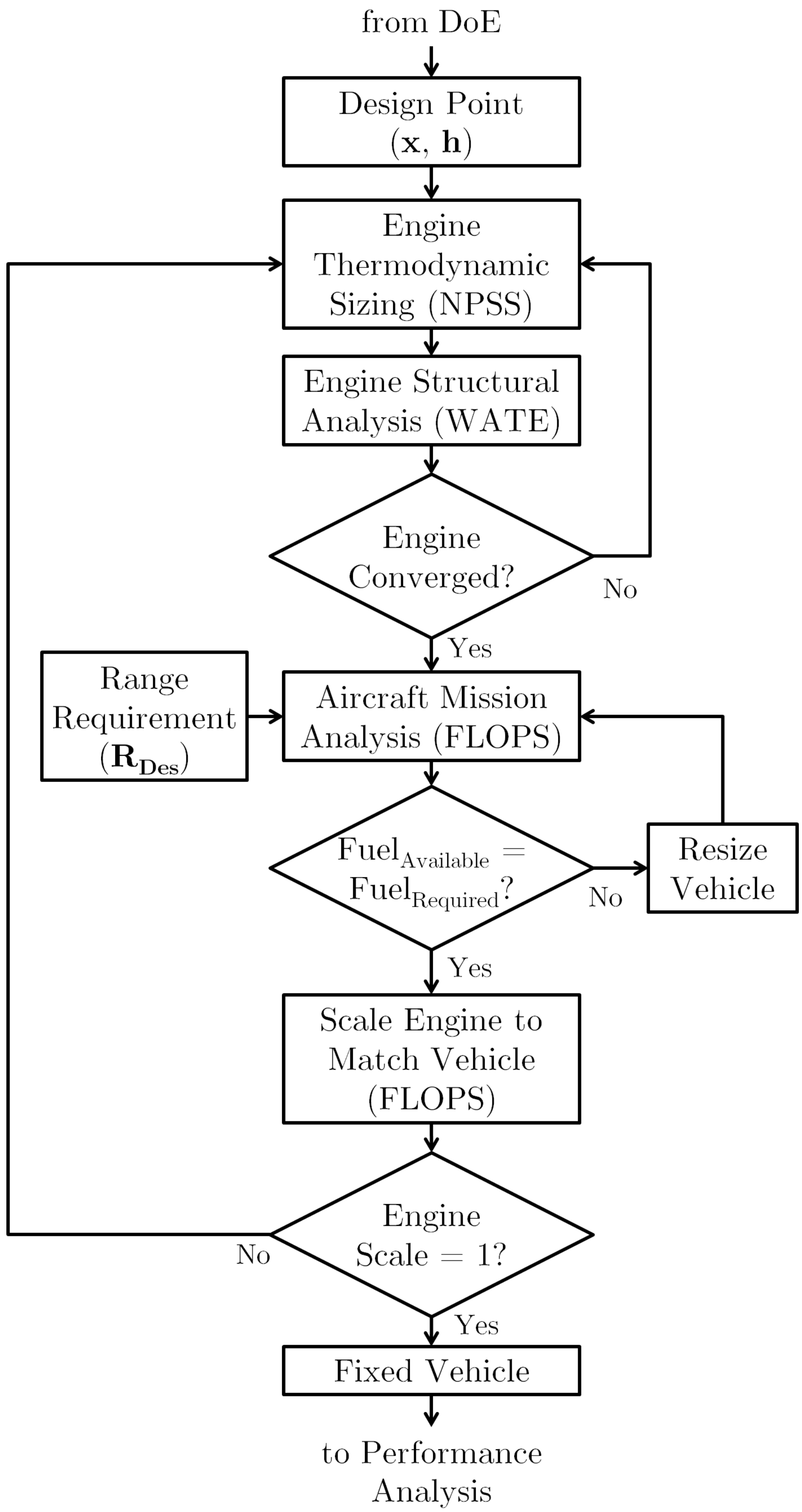


Figure 55: Model Sizing Logic

the thermodynamics of the engine have been determined, a structural analysis of the engine is performed by the Weight Analysis of Turbine Engines (WATE). A convergence check is performed to ensure that the two code outputs describe the same engine and that assumptions made during the thermodynamic analysis were supported by the turbomachinery analysis.

After a converged engine solution has been created, the aircraft mission analysis can be performed by the FLight OPtimization System (FLOPS). In this analysis the range requirement (R_{Des}) dictates the size of the vehicle, given the design point specified by the DoE. The aircraft mission analysis will attempt to match the fuel required for the design mission to the fuel available within the aircraft geometry. If these two values do not match, the aircraft code will continue to resize the gross weight and geometry of the vehicle until they are consistent.

After the aircraft converges to a given size, the engine is scaled to match. Since the thrust-to-weight ratio (T/W) is specified as a design input, the engine thrust will be scaled to a fixed ratio with respect to the final sized maximum takeoff gross weight ($MTOW$). Should the engine be scaled more than a small amount, the model will return to the engine thermodynamic analysis to reassess the new engine. If the engine thrust output from the mission analysis matches the thrust from the engine sizing, the vehicle is considered sized. This vehicle configuration is "fixed" in place from this point forward. The fixed vehicle is then output to the performance analysis.

The performance analysis within the model is used to generate the responses necessary to generate surrogate models. The process used to evaluate the vehicle performance is shown in Figure 56. The vehicle output from the sizing module is fixed, meaning that its dimensions and overall maximum weights no longer change. Next, the margins (h) implemented during sizing are removed from the vehicle. The rationalization behind this step is discussed in detail in Section 3.2.

Once the margins have been removed from the fixed vehicle, a specific uncertainty

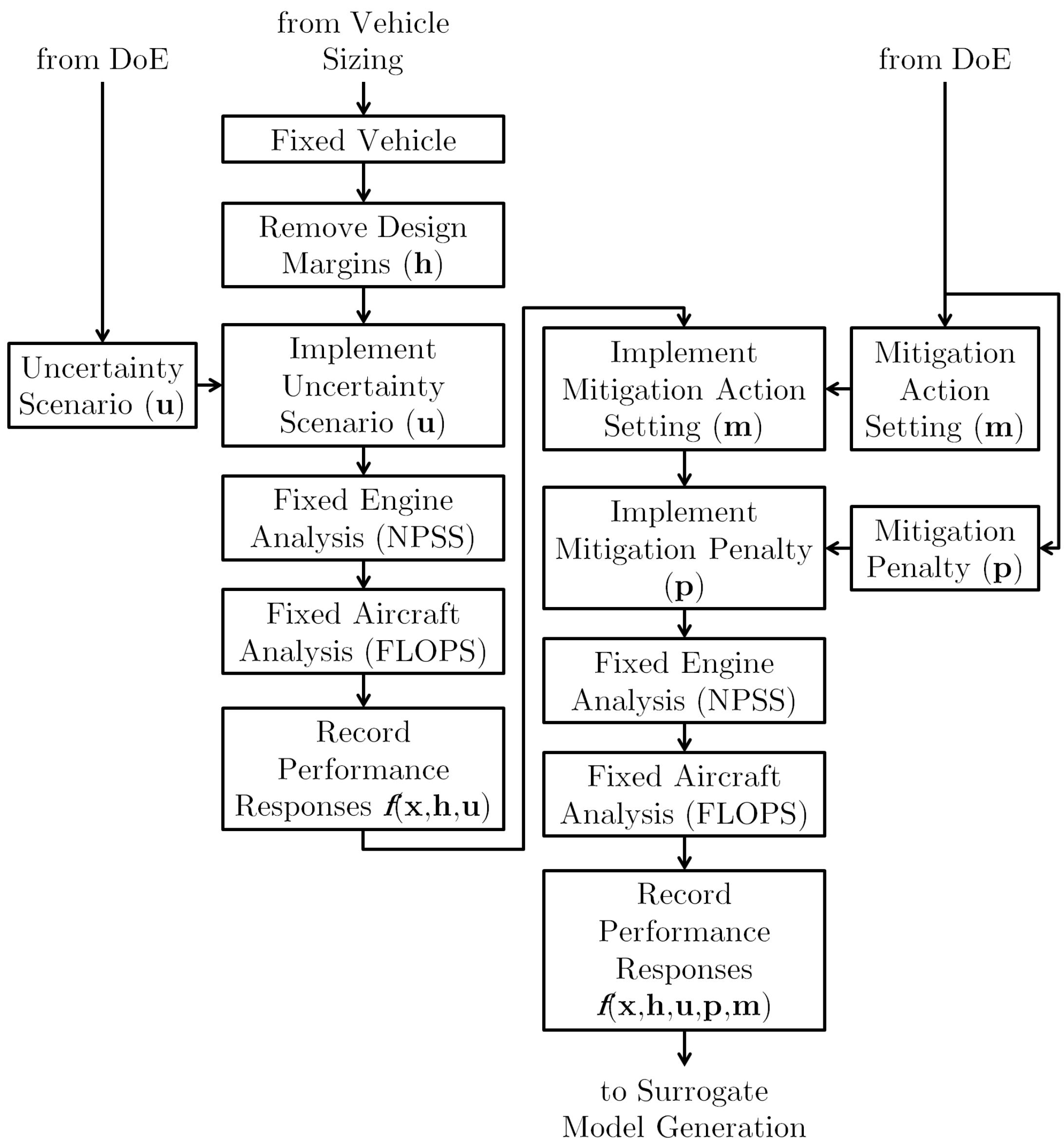


Figure 56: Model Performance Logic

scenario (u) is brought in from the DoE. This uncertainty scenario is implemented, altering the performance of the fixed vehicle from what was initially sized. A fixed engine analysis is then performed using NPSS and WATE. Once the new engine performance is determined, this information is given to the fixed aircraft analysis FLOPS model and the performance responses from the aircraft analysis are recorded. These responses will be functions of the design variables (x), the uncertainty margins (h), and the uncertainty scenario investigated (u).

Next, a level for each of the mitigation actions (m) is read from the Design of Experiments. This mitigation action setting is implemented in the fixed vehicle model. A level of mitigation penalty (p) also comes from the DoE, and this value is input into the vehicle model. The engine is run again in an off-design mode to calculate any changes in thrust and fuel flow performance that result. A fixed aircraft analysis is then executed to evaluate the response behavior after the mitigation actions have been implemented. Finally, the performance responses are recorded. These responses are functions of all input parameters to the model, including the mitigation actions and penalties.

6.8 Step 8: Create Surrogate Models

It was known a priori that at least one of the responses from the aircraft analysis code could not be fit sufficiently using a response surface equation. Thus, an Artificial Neural Network (ANN) described in Section 2.6.2 was selected as the surrogate model type to represent the behavior of the responses. This type of surrogate model is good at emulating responses which are non-linear with respect to the input variables. The neural networks were generated using the Basic Regression Analysis for Integrated Neural Networks (BRAINN) [42]. As such, a large space-filling design of experiments was needed. A normalized Design of Experiments was constructed using the variables and ranges defined in Sections 6.2 to 6.4. MATLAB's lhsdesign function was used to generate a Latin Hypercube design with 20,000 cases for the 17 variables. 5,0000 random cases were generated to allow for goodness of fit testing of the surrogate models. The normalized Design of Experiments was converted to the appropriate ranges for each of the design, uncertainty, and mitigation variables in a Microsoft Excel spreadsheet. This resulting DoE was fed into the deterministic aircraft analysis code described in Step 7.

Surrogate models were then generated using this data. These surrogate models are substituted into Figures 49 to 51 to replace all aircraft performance analyses with the resulting equations. Goodness of fit metrics were generated for each surrogate model and are included in Appendix B.

6.9 Step 9: Create Uncertainty Quantification & Management Environment

The uncertainty quantification and management environment described in Section 5.9 has been developed and implemented in a MATLAB coding environment. The actual code employed in these studies is included in Appendix C. As this environment constitutes the core of the ARMOUR methodology, it synthesizes information from all of the previous steps. The range of variability for the design variables and margins were established as indicated by Table 5.

Uncertainty variables were initialized and their ranges limited as indicated by Table 3. The uncertainty distributions elicited during Step 3 were then input into the environment. Samples were drawn from these distributions for analysis. For this study, a 4-by-10000 array of pseudo random numbers was generated in MATLAB. Since the uncertainty distributions used in this study were independent uniform distributions, this array was scaled to fit the upper and lower bounds of the uncertainty variables listed in Table 3. Other uncertainty distribution types including triangular, truncated normal, and correlated uniform were implemented but not used in any of the final analyses.

The range of mitigation actions was established based on Table 4. Additionally, a weight penalty function was constructed which penalized the aircraft for both the post-sizing fuel addition mitigation action and the augmented high-lift device mitigation action. The individual penalties for each mitigation action from Equations (45) and (47) were both assessed and added together to achieve the final penalty imposed on the design.

Once these basics had been established, a global optimizer was initialized which operated on the design variables and margins. For this study, a genetic algorithm intrinsic to MATLAB (ga(x)) was be used as the global optimizer. It attempted to find a design point which minimizes the expected block fuel for economic range as specified during Step 5. Also, the global optimizer was be constrained to meet or exceed all reliability objective established during step 5.

The aircraft performance was then evaluated via the surrogate models to assess the probability of compliance ($P\left(Compliance \mid x, h\right)$) of the fixed vehicle as well as the expected block fuel needs for an economic range mission ($\mathbb{E}[BF_{Econ}]$). The surrogate models developed in Step 8 were used in place of direct calls to the aircraft analysis code to speed up the analysis. The design and margin settings selected by the optimizer, along with the uncertainty distributions generated earlier, were then input into the surrogate models. MATLAB's element-wise math was used to evaluate all of the uncertainty scenarios simultaneously for the given design point. The results from the surrogate models were then compared to the performance constraints in Table 2 which were established in Step 2. Using Equation (17), a probability of compliance was calculated for the given design. The set of uncertainty scenarios which constitute the failed set ($u \in U_{Fail}$) were selected via Equation (15). This set of uncertainty scenarios were then given to the mitigation analysis.

The mitigation analysis took in the failed uncertainty scenarios ($u \in U_{Fail}$) and evaluated these scenarios for possible recovery through mitigation actions. A selected set of mitigation actions were created for evaluation. Multiple options were considered for this mitigation evaluation, as detailed in Section 7.4. For most of the results in Chapter 7, the "Hybrid (1D and Random)[1]" set of mitigation scenarios was employed.

With the set of mitigation scenarios defined, a for loop was established to evaluate each mitigation scenario in turn. For each mitigation scenario, the performance responses were evaluated simultaneously for all failed uncertainty scenarios ($u \in U_{Fail}$) using MATLAB's element-wise math. All performance metrics were compared to their respective constraints for each uncertainty scenario. Whenever a mitigation action caused an uncertainty scenario to comply with all constraints simultaneously, this scenario was considered recovered as defined by Equation (19).

Whenever a scenario was recovered, the economic range mission block fuel for that scenario was recorded. If no other mitigation action had yet recovered that scenario, this block fuel would be preserved for use in the final objective constraint calculation from Equation (25). This calculation has been modified into Equation (53) to consider the specific needs of this implementation. If another set of mitigation actions were successful in recovering the failed uncertainty scenario ($u \in U_{Fail}$), then the resulting block fuel from these two scenarios was compared and the lower of the two values was kept for use in the objective constraint calculation.

Once all samples were drawn, the overall probability of success was calculated using Equation (24). Also, the expected block fuel for the economic mission ($\mathbb{E}[BF(x)]$) was evaluated via Equation (53). These values were then returned to the optimizer.

$$\mathbb{E}[BF(x)] = \frac{\int_{u \in A} BF\left(x, h, u, 0\right) du + \int_{u \in A_M} BF\left(x, h, u, m\right) du}{\int_{u \in A} (1) du + \int_{u \in A_M} (1) du} \tag{53}$$

The expected block fuel required for the economic range mission ($\mathbb{E}[BF_{Econ}]$) was used as the objective function of the optimizer. This optimizer attempted to select a design (x, h) in order to minimize the expected block fuel. Again, the probability

[1]Defined in Section 7.4.3.

of success constraint ensured that the probabilistic performance did not fall below the specified threshold. The optimizer input new settings of design variables (x) and uncertainty margins (h) into the vehicle sizing algorithm to attempt to minimize the expected block fuel while maintaining a level of success. Once the optimum was found, the design point (x, h) was recorded and reported back to the user along with the final objective value and the full set of reliability measures.

6.10 Step 10: Execute UQ&M Environment

Executing the Uncertainty Quantification and Management environment is as simple as running a MATLAB script and waiting for the code to finish. The resulting design(s) will meet or exceed the reliability targets specified in Step 5, within a tolerance established in the optimizer created during Step 9. For the studies located in Section 7.7, the UQ&M environment was modified to take in an array of reliability targets and return a set of designs which met each target in sequence. This allowed for the construction of a set of designs, each with different reliability targets and expected block fuels, from which a designer could select the aircraft deemed most appropriate for further development.

6.11 Chapter Summary

In this chapter, an implementation of the step-by-step ARMOUR methodology described in Chapter 5 was adapted to fit the specific needs of the design of a large civil transport aircraft. Variables, constraints, and mitigation actions were discussed along with applicable ranges. The design objectives were selected to constrain the resulting optimized solution. Existing physics-based design tools formed the foundation of this analysis. An uncertainty quantification and management environment was constructed in MATLAB using surrogates of the aircraft sizing and performance analyses. This environment will be exercised in Chapter 7 to glean additional information about the behavior of the aircraft design process under uncertainty.

CHAPTER VII

RESULTS

Now that the methodology developed in Chapter 5 has been instantiated using the specific implementation from Chapter 6, a number of studies can be performed to glean insight into an aircraft design process which considers design uncertainty with recovery through mitigation actions. The first two studies in Sections 7.2 and 7.3 investigate the impact of uncertainty margins on the two of the reliability measures used in this work, probability of compliance and probability of recovery. Section 7.5 explores the impact of penalties imposed on mitigation actions on the probability of recovery for designs. The design and margin space is explored in Section 7.6. The results of the design space exploration are compared to a set of optimized Pareto frontiers in Section 7.7.

7.1 Margin Implementation Testing

Section 3.2 discussed a desire to quantitatively set margins to achieve a probability of compliance in Research Question 2.

> **Research Question 2** *Is it possible to select a desired probability of compliance and then quantitatively determine a level of margin which will yield that probability of compliance?*

Previous uncertainty quantification methods have implemented performance target setting, which is conceptually similar to uncertainty margins. Thus, it was assumed that uncertainty margins could be implemented in a probabilistic framework. This implementation would determine the probability of compliance based on a setting of uncertainty margins. Further, this framework should allow for the selection of uncertainty margins based on a particular desired level of probability of compliance. This concept was formalized in Hypothesis 2.

> **Hypothesis 2** *By including an uncertainty margin during the sizing process and removing it (but not its effect) before the uncertainty analysis, the impact of margin on the probability of compliance can be seen. Using an optimizer, it will be straight-forward to determine an appropriate level of margin to achieve a desired probability of compliance.*

A partial form of the final implementation established in Chapter 6 is used to test the impact of assessing uncertainty margins in an uncertainty quantification environment. Using this partial implementation, a baseline vehicle was assessed to determine its expected performance and probability of compliance. This baseline vector and its performance is tabulated in Table 8.

Table 8: Margin Optimization Baseline

Parameter	Baseline
TWR	0.2962
WSR	133.34
AR	10.0
LapseRate	0.2014
FPR	1.50
OPR	42.58
h_{EW}	0.030
h_{Drag}	1.030
$h_{FuelFlow}$	1.030
Compliance	71.5%
BF_{Det}	227,834
$E[BF_{Des}]$	223,478
$E[BF_{Econ}]$	115,526

Once the baseline performance is established, the levels of margins were adjusted, leaving all other design variables constant. Uncertainty margins input sweeping from no margin up to seven percent margin over the code-predicted values. All three margins were kept at equal levels, as they were changed. Figure 57 shows the probability of compliance for the baseline design at different settings of uncertainty margins. Along the x-axis is the margin value assigned to all three uncertainty margins. The blue line corresponds to the resulting probability of compliance on the y-axis, resulting from the reliability assessment.

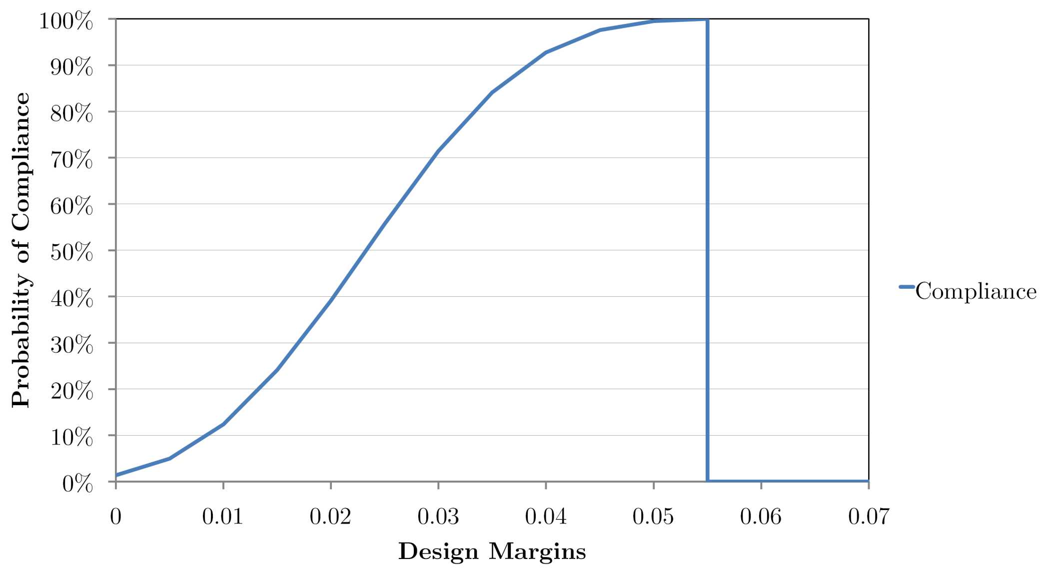


Figure 57: Baseline Compliance versus Margin

Using an optimizer to adjust all three uncertainty margins simultaneously indicates that to achieve an 80 percent probability of compliance ($P\left(Compliance \mid x, h\right)$) with equal margin values, uncertainty margins should be set at 3.336 percent. This will result in roughly 550 lbs increase in the expected economic range block fuel ($E[BF_{Econ}]$).

This conclusion already meets the minimum goals necessary to support Hypothesis 2. However, much more can be accomplished by including margins in the uncertainty quantification environment.

Using the established margin, an optimizer was used to select the best deterministic design, as might be done in a standard conceptual design study. The optimizer was allowed to select any combination of design variables with the goal of achieving a minimum deterministic economic range block fuel for that design. Analyzing this new design through the uncertainty quantification environment showed that the new deterministic design had improved the expected economic range block fuel by 8,360 lbs, but the probability of compliance had dropped to 48.4 percent.

Using this new deterministic design, the optimizer was again used, allowing uncertainty margins only to change. This optimizer was given access to the reliability analysis and asked to minimize expected block fuel while achieving a probability of compliance of 80 percent. This new "Opt Margin" design based on the deterministic optimization design was able to meet the probability of compliance goal by increasing uncertainty margins, but it increased the expected economic block fuel by 2,200 lbs.

As a final point of comparison, the optimizer was allowed to vary both design variables and uncertainty margins. The optimizer attempted to find the lowest expected block fuel while still being restricted to a resulting design with 80 percent probability of compliance. This final "Full Optimization" design achieved the lowest expected block fuel while still meeting the probability of compliance goal. A summary of these results and the corresponding design vectors is contained within Table 9.

Table 9: Margin Optimization Test Results

Parameter	Baseline	Deterministic	Opt Margin	Full Optimization
TWR	0.2962	0.2686	0.2686	0.2638
WSR	133.34	137.71	137.71	135.09
AR	10.0	10.79	10.79	11.00
LapseRate	0.2014	0.206	0.206	0.2089
FPR	1.50	1.5198	1.5198	1.5625
OPR	42.58	46.635	46.635	46.838
h_{EW}	0.030	0.034	0.0446	0.0225
h_{Drag}	1.030	1.034	1.0427	1.0447
$h_{FuelFlow}$	1.030	1.034	1.0457	1.0428
Compliance	71.5%	48.4%	80.0%	81.3%
BF_{Det}	227,834	214,892	223,527	215,870
$E[BF_{Des}]$	223,478	207,704	210,212	205,714
$E[BF_{Econ}]$	115,526	107,721	109,389	106,727

7.2 *Relationship between Margin and Compliance*

It would be easy to assume that, in general, increasing uncertainty margins would lead to an increased chance that the resulting design would be compliant with all imposed constraints. Using the methodology as implemented in Chapter 6, this idea can be investigated to test how true it may be. To test this concept, margins will be varied for individual designs to investigate the impact on the probability of compliance. Further, a design space exploration will be performed to investigate whether overall trends exist for the entire design space.

7.2.1 Investigating Individual Designs

Table 10 demonstrates the individual designs investigated in Figures 58 and 59. These designs differ from each other, greatly for some design variables, but they demonstrate similar levels of performance when viewed from a high level. Of particular note is that the designs have very different levels of uncertainty margin for all three variables (h_{EW}, h_{Drag}, and $h_{FuelFlow}$). Despite these differences, the designs have a similar probabilities of compliance and expected block fuel performance.

Table 10: Designs to Investigate Margin vs. Compliance

Parameter	Design 1	Design 2
TWR	0.3232	0.2949
WSR	139.23	128.58
AR	9.515	9.555
LapseRate	0.1903	0.1951
FPR	1.471	1.495
OPR	46.65	42.58
h_{EW}	1.001	1.033
h_{Drag}	1.051	1.030
$h_{FuelFlow}$	1.069	1.064
Compliance	81.7%	83.7%
$\mathbb{E}[BF_{Des}]$	229,954	230,426
$\mathbb{E}[BF_{Econ}]$	118,524	118,807

The trend of probability of compliance with respect to individual margin changes were investigated for each design individually. Each design investigated was based off either Design 1 or Design 2 from Table 10. A sweep of each uncertainty margin (h_{EW}, h_{Drag}, and $h_{FuelFlow}$) was constructed, varying each margin from as low as no change from the deterministic code prediction up to a 10% increase over the deterministic prediction. These resulting designs were input into the the aircraft assessment implementation developed in Chapter 6 without any optimization in order to evaluate the performance of each individual design. The probability of compliance for each resulting design was recorded.

Figure 58 shows the impact of changing individual margins for Design 1. Using this design as a baseline, only one margin variable is changed at a time while all other variables are left at the setting from Table 10. Each assessment performs a sizing analysis to determine the new dimensions of the resulting aircraft. Following this, the frozen aircraft is assessed for a large number of uncertainty scenarios to accurately determine the probability of success. The three different lines all represent a set of designs, each with a different margin being changed. These margins are swept from no margin (1.0) to ten percent margin (1.1) on the x-axis. The y-axis shows the corresponding probability of compliance for the resulting aircraft. For each line, the baseline setting of the margin for Design 1 are shown by the symbol corresponding symbol. It should be apparent that all three symbols correspond to the same probability of compliance as Design 1 in Table 10. The empty weight margin (h_{EW}) is shown in blue. Increasing values of this margin cause the probability of compliance to monotonically increase from 81.7 percent compliance to full compliance with all constraints. The red line shows the impact of varying the drag margin (h_{Drag}). Again, increasing values of this margin cause a monotonic increase in the probability of compliance, this time from about 35 percent to 99 percent. The green line shows similar trends for the engine fuel flow margin ($h_{FuelFlow}$). Increasing values of this margin also show a monotonic increase from a very low 13 percent up to 95 percent.

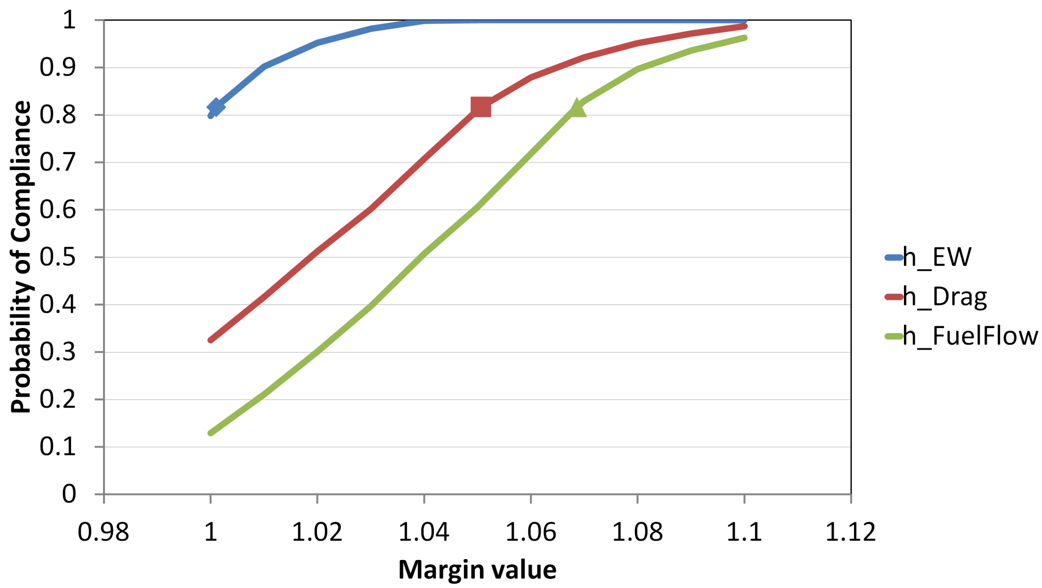


Figure 58: Probability of Compliance vs. Margin for Design 1

For Design 1, increasing margins lead to an overall increase in the probability of compliance. The primary active constraint for this design was range, so the result that increasing margin will improve the probability of compliance would be expected.

Figure 59 shows the behavior of varying uncertainty margins for Design 2 from Table 10. Just as in Figure 58, a sweep was performed for each margin separately, varying values from no margin to ten percent margin along the x-axis. The corresponding probability of compliance for each design is shown on the y-axis. The three margins, empty weight margin (h_{EW}), drag margin (h_{Drag}), and engine fuel flow margin ($h_{FuelFlow}$), are shown in blue, red, and green, respectively. Their baseline values from Design 2 are indicated by their respective symbol, and also all correspond to Design 2's probability of compliance of 83.7 percent. For low values of margins, it once again appears that increasing margin leads to increased probability of compliance; however, this trend does not continue indefinitely. For both empty weight margin and drag margin, a point is reached at which the probability of compliance no longer improves. Instead, the probability of compliance drops instantaneously to zero. For empty weight margin (h_{EW}) this occurs at about seven percent margin. For drag

(h_{Drag}) this precipitous drop occurs at nine percent margin.

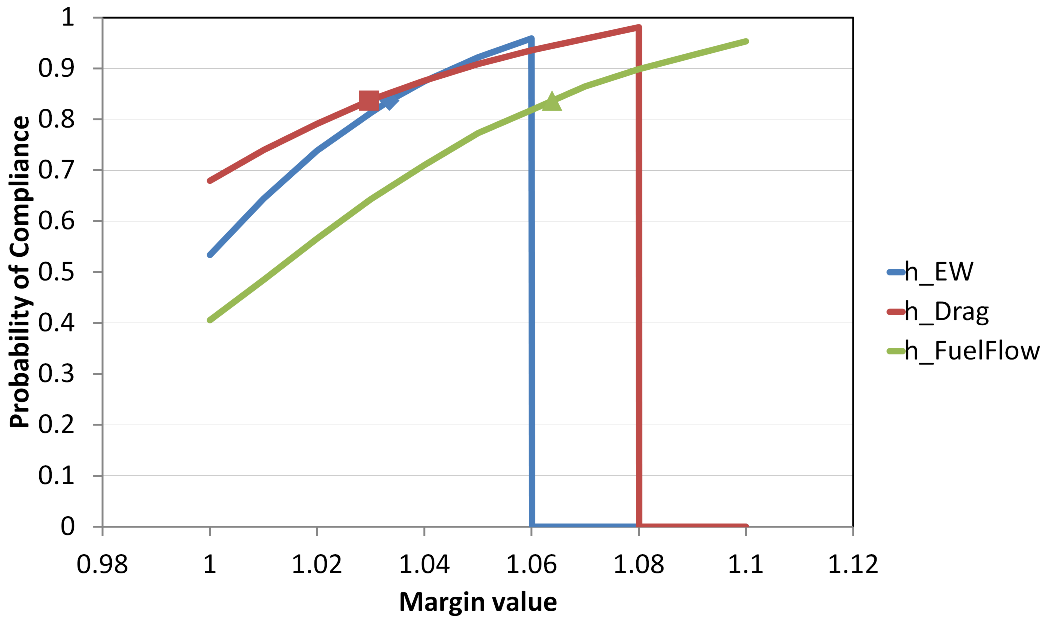


Figure 59: Probability of Compliance vs. Margin for Design 2

Looking at detailed results of the analysis, it can be seen that a wing span constraint is becoming active. The reason that compliance drops instantaneously rather than being a smooth slope as with the rise of compliance for lower margin values is that span is fixed after the sizing loop. Because span is fixed before any knowledge is gained about the uncertainty variables, it cannot be affected by changes in their values. Thus, there is no uncertainty associated with the aircraft's compliance with that constraint. The probability of compliance simply drops to zero after the aircraft reaches a certain size. Because the aircraft is sized with these margins imposed, as the margins increase so does the maximum takeoff weight and size of the vehicle. Since the wing loading (WSR) is being held constant for a design, the wing area increases as margins increase. This increased wing area leads to an increased in wing span.

A conclusion which could be drawn already from these two examples is that higher levels of margin do not always equate to a higher probability of compliance with performance constraints. In this manner the relationship between margins and performance metrics mimics that of any other design variable.

7.2.2 Design Space Exploration

Rather than continuing to look at many individual designs separately, it will be more helpful to try to examine a large portion of the design space. To enable this visualization, an exploration of all design variables and uncertainty margins was performed via Monte Carlo Simulation, generating 50,000 different designs. For each resulting design, the process described in Chapter 6 was executed to determine the probability of compliance for the design.

Examining the data from this large exploration of designs simultaneously proved difficult. Instead, designs were binned based on their three margin values. The designs were separated into a total of 125 bins – 5 for each independent margin dimension (h_{EW}, h_{Drag}, and $h_{FuelFlow}$). With the bins established and the designs sorted, an average was taken of the probabilities of compliance seen across the designs within each bin. This average compliance was then plotted against each of the uncertainty margins.

Figure 60 shows the results for one of these uncertainty margin bins. This chart plots the average probability of compliance on the y-axis versus the empty weight margin (h_{EW}) on the x-axis, varying between no margin and ten percent margin. The chart only shows roughly 2,000 of the 50,000 designs – the ones which have both a drag margin between zero and two percent as well as a fuel flow margin between zero and two percent. Each dot represents a bin of roughly 400 different designs which all have about the same values individually for each of the three uncertainty margins. For each bin, the probability of compliance was averaged across all designs. This average probability of compliance for each bin is plotted as a dot on the chart, located horizontally at its empty weight margin bin. A curve is fit through these data points to aid the viewer's eye in seeing trends. The result is an ability to see the average behavior of probability of compliance versus the empty weight margin for a particular bin of drag and fuel flow margins.

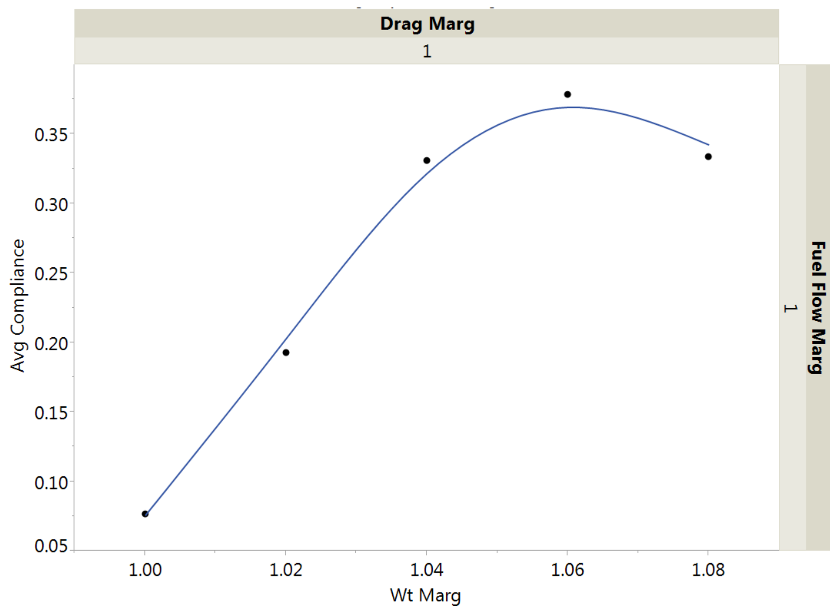


Figure 60: Average Compliance vs. Empty Weight Margin

Examining each combination of these plots individually would take significant time and energy. In an attempt to look at the entire space simultaneously, individual plots like Figure 60 are shown as a large grid of plots in Figure 61. This resulting plot gives the ability to examine the average compliance versus all three margins simultaneously. Just as with the Figure 60, each subplot of Figure 61 represents a subset of about 2,000 of the 50,000 designs which correspond to a combination of drag margin (h_{EW}) and fuel flow margin ($h_{FuelFlow}$). The first plot in the top left only contains designs which have both a drag margin between zero and two percent as well as a fuel flow margin between zero and two percent, just like Figure 60. Moving to the right, each plot shows a different 2,000 designs which have progressively two percent higher drag than the one to its left. The designs in subplot all the way to the right on the top has a drag margin between eight and ten percent and a fuel flow margin between zero and two percent. Going down, each progressive subplot has 2,000 different designs with about two percent more drag margin than the plot above it. Thus, the subplot on the bottom right has a fuel flow margin between eight percent and ten percent as well as a drag margin between eight percent and ten percent. Each individual plot is identical in implementation to Figure 60. In it, each dot represents the average behavior of about 400 designs, each of which have the same empty weight margin, drag margin, and fuel flow margin. The empty weight margin is plotted on the x-axis versus the average probability of compliance for this set of designs on the y-axis.

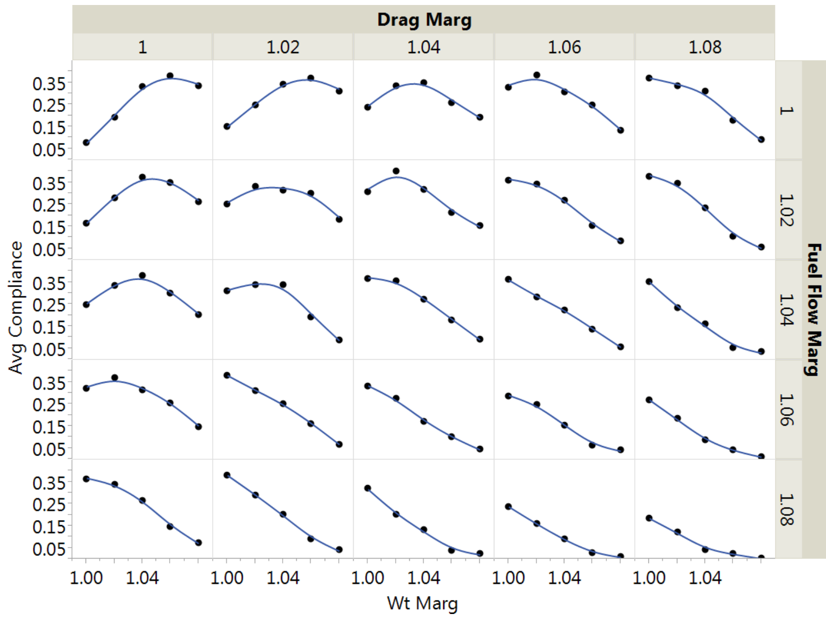


Figure 61: Average Compliance vs. Uncertainty Margins

Examining Figure 61 leads to the following generalizations. When all margins are small, increasing any margin tends to improve the average probability of compliance of the design space. However, there appears to be a limit to this behavior which causes margins to lose effectiveness. Indeed, when any margin or combination of margins are at a high value, further increasing margin tends to degrade the probability of compliance of the design space. Neither of these statements are surprising considering the contrast in single design examples shown in Section 7.2.1. However, the knowledge that those trends would generally hold for the design space could not be deduced from merely two designs.

7.3 Relationship between Margin and Recovery

It will also be beneficial to examine whether there exists a relationship between the probability of recovery and uncertainty margins. As with the probability of compliance, it will be helpful to examine the design space. Since the requirements were similar, the same 50,000 point Monte Carlo simulation of designs was used to construct this evaluation. The probability of recovery was estimated using 250 random search mitigation actions for each design.

The designs were once again binned based on their margins. The designs were separated into a total of 125 bins – 5 for each independent margin dimension (h_{EW}, h_{Drag}, and $h_{FuelFlow}$), corresponding to two percent increments. With the bins established and the designs sorted, an average was taken of the probabilities of recovery seen across the designs within each bin. This average recovery is plotted against each of the uncertainty margins.

To examine the average probability of recovery versus all three uncertainty margins for the entire design space, Figure 62 was created. This resulting plot gives the ability to examine the average recovery versus all three margins simultaneously. Just as with the Figure 61, each subplot of Figure 62 represents a subset of about 2,000 of the 50,000 designs which correspond to a combination of drag margin (h_{EW}) and fuel flow margin ($h_{FuelFlow}$). The first plot in the top left only contains designs which have both a drag margin between zero and two percent as well as a fuel flow margin between zero and two percent. Moving to the right, each plot shows a different 2,000 designs which have progressively two percent higher drag than the one to its left. The designs in subplot all the way to the right on the top has a drag margin between eight percent and ten percent and a fuel flow margin between zero and two percent. Going down, each progressive subplot has 2,000 different designs with about two percent more drag margin than the plot above it. Thus, the subplot on the bottom right has a fuel flow margin between eight percent and ten percent as well as a drag margin between eight percent and ten percent. Each individual plot is identical in implementation. In a plot, each dot represents the average probability of recovery behavior of about 400 designs, each of which have the same empty weight margin, drag margin, and fuel flow margin. The empty weight margin is plotted on the x-axis versus the average probability of recovery for this set of designs on the y-axis.

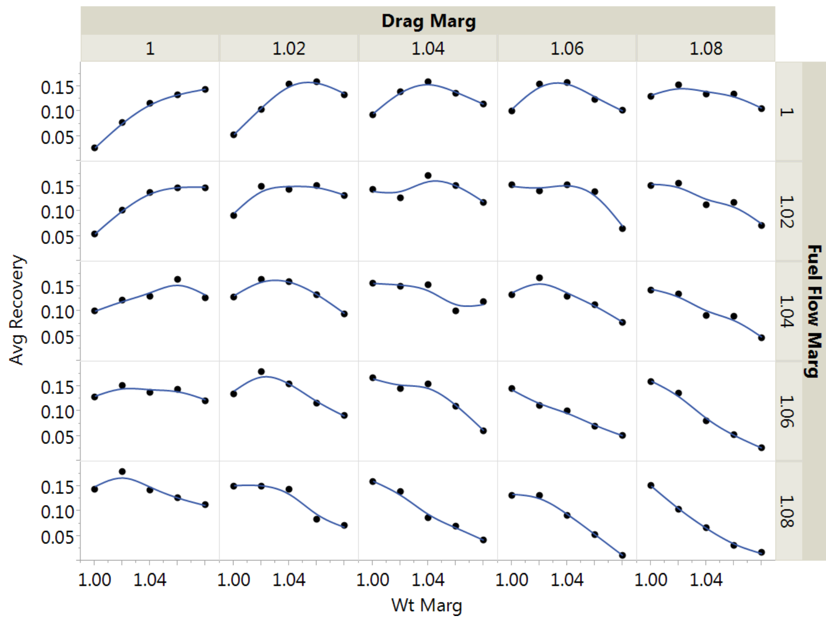


Figure 62: Average Recovery vs. Uncertainty Margins

Examining Figure 62 reveals similar trends to those shown when investigating the average probability of compliance. It appears the increasing margins improve the probability of recovery when margins are low, and that they degrade the probability of recovery when margins are high. However, it is possible that this effect is at least in part due to the effect of these trends existing in the compliance space.

To determine how much of the effects seen in Figure 62 are a result of the probability of compliance, all designs with a probability compliance of less than 50 percent were removed. This left about 11,000 designs remaining of the original 50,000. This data was binned like before and is plotted in Figure 63. Each data point is still the result of the designs which fit into a bin corresponding to a combination of drag margin (h_{EW}) and fuel flow margin ($h_{FuelFlow}$). The first plot in the top left only contains designs which have both a drag fuel flow margins between zero and two percent. Moving to the right, each plot shows a different set of designs which have progressively two percent higher drag than the one to its left. Going down, each progressive subplot contains different designs with about two percent more drag margin than the plot above it. In addition to the dimensions present before, a "hot body" color scale has been included which indicates the number of cases which occur in each bin going from yellow to red to black. The darker the data point, the more designs which exist in that margin bin which met the 50 percent probability of compliance limit. Each dot represents as few as 1 design in yellow and as many as 200 different designs in black. Each dot still represents the average probability of recovery of the remaining designs, each of which have the same empty weight margin, drag margin, and fuel flow margin as well as a probability of compliance greater than 50 percent. The empty weight margin is plotted on the x-axis versus the average probability of recovery for this set of designs on the y-axis.

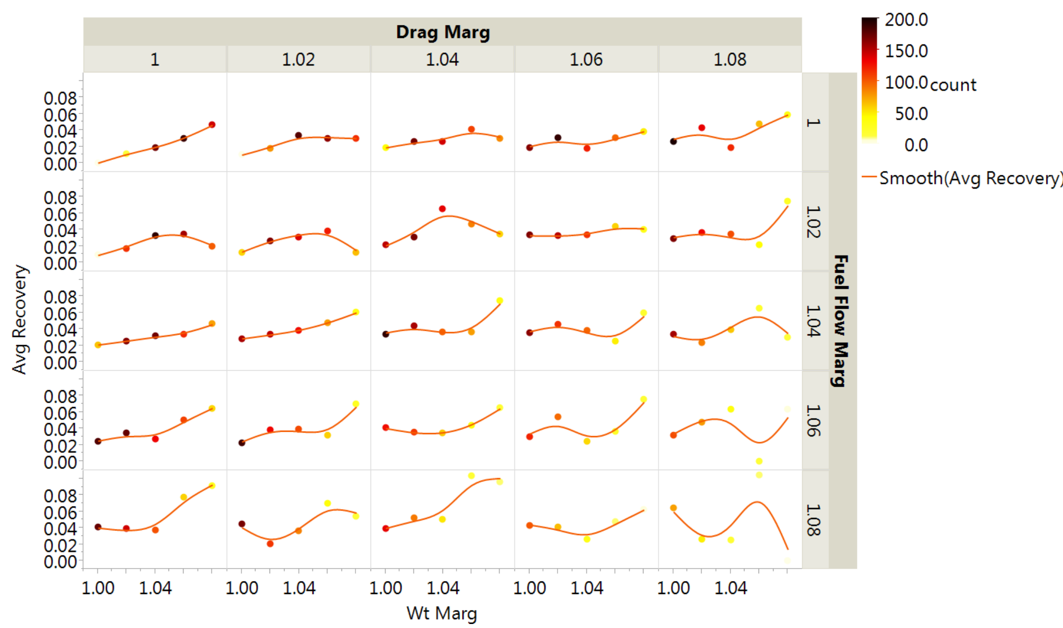


Figure 63: Average Recovery vs. Uncertainty Margins > 50% $P\,(Compliance \mid x, h)$

Figure 63 illustrates that, in general, increasing margins remain effective at improving the probability of recovery at higher values of margin than before. This indicates that IF the increase in a margin does not cause the design to decrease in compliance due to a constraint becoming active, the average probability of recovery will increase with increasing margin.

There still appears to be a drop off to this trend at very high margins. However, the trends become volatile due to the lack of observable designs which met the probability of compliance limit in those regions. The lack of data makes it difficult to make concrete conclusions about the trend of probability of recovery with respect to uncertainty margins for extremely high values of those uncertainty margins. This may not be of particular concern, since the results Section 7.2.1 indicated that the probability of compliance is generally low in this region. Coupled with the fact that excessive margins lead to larger, higher-weight aircraft, this implies that this region is likely not of particular interest to a designer, anyways.

7.4 Method for Mitigation Space Exploration

Now that Hypothesis 3 has been supported, it will be helpful to investigate what method should be used to determine the appropriate level of mitigation for each failed uncertainty scenario. This assessment will need to accurately determine the probability of recovery for a given design by investigating the mitigation options for different uncertainty scenarios.

Recall that this part of the analysis occurs very late within the methodology. Because this entire analysis will need to be called separately for each uncertainty scenario, the number of function calls to the constraint evaluations at this level has a very large impact on the total number of function calls required. Thus, even an efficient optimizer implemented here could balloon the required analysis time to a prohibitive level, rendering the entire methodology impractical. Rather than simply implementing an optimizer to find a solution separately for each uncertainty scenario, it would be beneficial to examine the requirements of the mitigation space and determine if an optimizer is even required.

The primary concern at this level is to determine the probability of recovery. Thus, it may not be necessary to minimize the level of mitigation applied. Indeed, all that is required is a determination of whether a combination of mitigation actions exists that can bring a failed uncertain scenario back into compliance.

It is unhelpful and somewhat nonsensical to try to define a "level of mitigation action" at the design level. Each successful recovery through mitigation actions may have a specific "level" associated with it; however, this is specific to not only the design but also to the specific uncertainty scenario under investigation. Since a breadth of uncertainty scenarios are assessed for each design, there will also be a breadth of mitigation levels associated with the given design. For some designs it may even be possible to have multiple active constraints, changing with the uncertainty scenarios, which will lead to completely different mitigation actions being used to recover the vehicle. Additionally, the mitigation actions are only used after a design has failed due to a specific uncertainty scenario.

Some information which is helpful to a designer is the likelihood of whether a given design can be recovered if it does fail. During the Conceptual Design phase, the aircraft gross design parameters are being selected. The mitigation actions will be used if and only if the inherent uncertainty in the design process causes the aircraft to fail to meet its goals. Whether the mitigation action will even be needed will not be known until the scenario specific to this aircraft is known, which will occur well after the Conceptual Design phase is completed.

Instead, by examining all possible scenarios, some insight can be gained into the probabilities of events occurring for each design. This information of this broad behavior across the uncertainty space will be helpful in deciding which design to select. The results of any specific scenario are less useful at this stage when compared to this more general information.

Planning for specific failures is less useful than knowing how likely it is that a design will comply with all performance constraints and how likely it is that the design can be recovered through the available mitigation actions.

Additionally, detailed knowledge of the "mitigation level" may be dangerous in the wrong hands. The concept of a mitigation action is to fix a design if and only if it is going to fail. Mitigation actions are by their very nature less desirable than simply changing the selected design point. Indeed, if they were more efficient than a design variable, they should be used as such.

Having this knowledge before it is necessary and without appropriate context may lead to an attempt to pre-mitigate a design, which is not intended. Implementing a mitigation action during Conceptual Design would simply be another way of re-designing the vehicle. Since mitigation actions are expected to be less efficient than selecting a new design point, this would degrade the design, potentially unnecessarily. This may also have some unintended consequences, since it is not the process which the proposed method models.

Instead, this information is intended to inform the design team. While this method is bringing probabilistic information forward from the Preliminary Design stage, it cannot change the fact that the decisions being made are during Conceptual Design. Indeed, the team need not make Preliminary Design decisions at this stage. Instead, whether or not to mitigate can wait until more information is known later on in the design process.

Additionally, the minimum amount of mitigation action will always be very near zero. For any design which has some non-zero and non-unity probability of compliance, there will be some uncertain scenarios under which the design will fail to meet some constraints and some other scenarios under which the design does meet all constraints. At the boundary between these regions, scenarios will exist where the design just barely fails to meet a design constraint. In this situation, the design will be so near to meeting the constraints that the amount of mitigation action will be trivial. Thus, quantifying a minimum amount of mitigation would be trivial and unhelpful information for the designer.

To determine the actual recovery of a design, a "truth model" was established. In this mode, 10,000 random search mitigation points were used to acquire a very accurate estimate of the design's probability of recovery. All other recovery estimates will be compared to this "truth model" in order to determine their accuracy at estimating the probability of recovery.

7.4.1 Recovery via Random Search

Another viable method to find the probability for investigating the mitigation space would be to use a random search. This method has already been shown to be significantly faster and about as accurate as employing an optimizer. If a random search is to be used for selecting mitigation actions, a number of search cases must be selected. To decrease the required run time, a minimum number of search points will be desired. Conversely, a large number of points may be necessary in order to correctly estimate the probability of recovery. In order to determine the appropriate number of cases which balance these two competing goals, a series of tests will be formulated to evaluate both the accuracy of the recovery estimate and the execution time required.

A large number of uncertainty scenarios will be used to ensure that the uncertainty space was fully explored. To ensure that the different number of mitigation search points can be compared fairly, the same uncertainty scenarios will be used for each design. This is accomplished by giving MATLAB's random number generator the same seed immediately before the uncertainty scenarios are generated. Following the generation of all scenarios, the random number generator is reset to a seed based on the computer's internal clock.

The impact of the number of random mitigation search points on the probability of recovery estimate was first investigated using individual designs. Designs were randomly selected after screening for those which had both some non-zero probability of compliance and some non-zero probability of recovery. Table 11 shows two example designs which will be initially evaluated to help determine an appropriate number of random search mitigation cases.

Table 11: Designs to Assess Mitigation Accuracy

Parameter	Design 1	Design 2
TWR	0.3355	0.2798
WSR	138.33	138.51
AR	10.44	10.33
h_{EW}	1.032	1.002
h_{Drag}	1.058	1.092
$h_{FuelFlow}$	1.006	1.021
LapseRate	0.182	0.206
FPR	1.486	1.447
OPR	44.65	46.72
$P\left(Compliance\right)$	64.59%	50.29%
$P\left(Success\right)$	72.81%	75.86%
$P\left(Recovery\right)$	8.22%	25.56%

To evaluate how many random search points are necessary to accurately estimate the probability of recovery, the methodology was executed independently multiple times. Each evaluation would be executed with a different number of random search cases. These evaluations would be compared against each other to determine the best combination of accuracy and speed. Since the number of random search cases required was unknown, a large range was established between as few as five (5) and as many as one thousand (1,000) random search cases. Since a random search operates on the same principle as a Monte Carlo simulation described in Section 2.4.1, the amount of error in the recovery estimate was expected to decrease with the square root of the number of cases. Thus, running numbers of cases in linear increments seemed inappropriate. To cover the gambit of cases, a hand-made log-like scale was implemented. The number of random cases executed for the following run of the methodology would either double (e.g. from 5 to 10) or multiply by two-and-a-half (e.g. from 100 to 250) from the number of cases random search mitigation cases previously executed.

Table 12: Number of Random Search Mitigation Cases to Execute

Mitigation Search Method		Run 1	2	3	4	5	6	7	8
Random	Cases	5	10	25	50	100	250	500	1000

Using this set of runs, the methodology was executed to determine the estimated probability of recovery under each set of mitigation search cases. Figure 64 shows the rate of convergence of the recovery estimate towards the probability of recovery predicted by the truth model for Design 1 from Table 11. On the y-axis is the predicted probability of recovery for a particular execution. On the x-axis is the number of random mitigation search cases used to acquire the estimate. The blue line shows the trend of the recovery estimate compared to the red dotted line which was the recovery found in the truth model. This figure shows that the probability of recovery converges to within half a percent of the actual value by 50 random search mitigation cases.

Figure 65 shows the results of this set of random mitigation search executions on Design 2. The number of random mitigation search cases from Table 12 is shown on the x-axis versus the resulting probability of recovery estimate on the y-axis. The blue line represents the recovery estimate for Design 2 while the red dotted line shows the recovery seen in the truth model. This figure shows that the probability of recovery converges to within half a percent of the actual value by 100 random search mitigation cases.

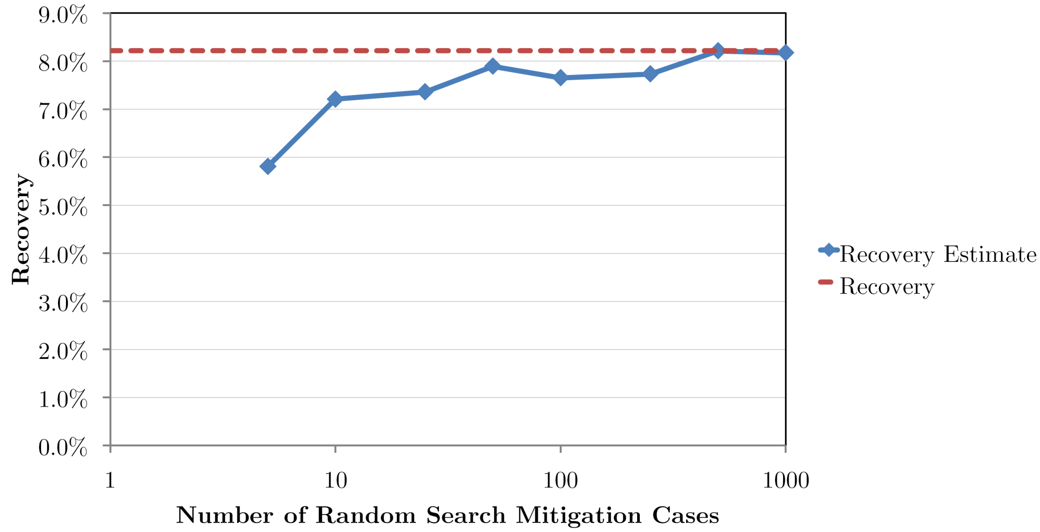


Figure 64: Recovery Estimate vs. Number of Mitigation Search Cases - Design 1

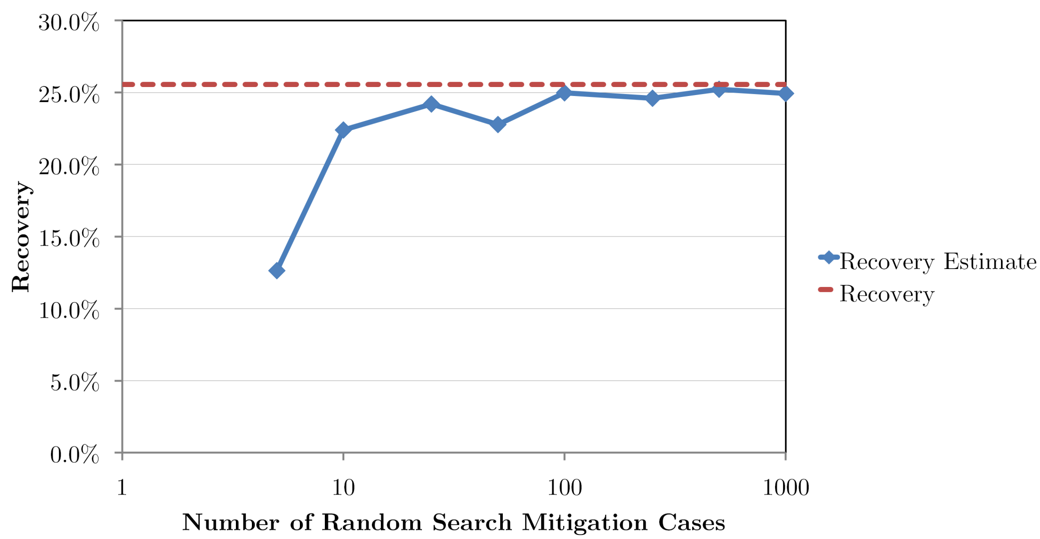


Figure 65: Recovery Estimate vs. Number of Mitigation Search Cases - Design 2

Examining the rate of convergence of the probability of recovery estimate for individual designs will quickly become tedious. Instead, it would be helpful to examine the design space. Statistics can be generated based on the number of random search cases to determine an appropriate number of mitigation cases to use for further analyses. A common way of graphically displaying statistical data about distributions is a box-and-whisker plot or box plot. The left side of Figure 66 shows multiple box plots. These box plots give an idea of the overall statistics of the distribution in a quickly readible manner.

In order to compare the different mitigation space exploration techniques, later charts contrast multiple *sets* of box plots. Large numbers of box plots on a single chart quickly become challenging to interpret; thus, the traditional box plots are simplified in order to compare across these sets. This simplification is shown in Figure 66. The left and right sides of the diagram contain the exact same informaiton about the set of five distributions. The left side shows five standard box plots. A line connects their median values of these five distributions. On the right, the boxes indicating the high quantile, median, and low quantile have been removed. Instead, a marker is used to indicate the location of the median. The whiskers remain, which allows the reader to determine the location of the quantiles along with the ends of the distribution. To visually connect the dataset, a lightly-colored region is used which connects the ends of the different distributions.

A large number of designs were generated using randomly-selected values of all design variables and design margins. Of these cases, a set of designs were selected by filtering out all cases which had a zero probability of recovery. Of the remaining cases, a set of one thousand designs were randomly selected to be used in this analysis. These designs were assessed using a random search throughout the mitigation space to determine their probability of recovery. Each design was run at each of the conditions described in Table 12. For each run, the estimated probability of recovery for a

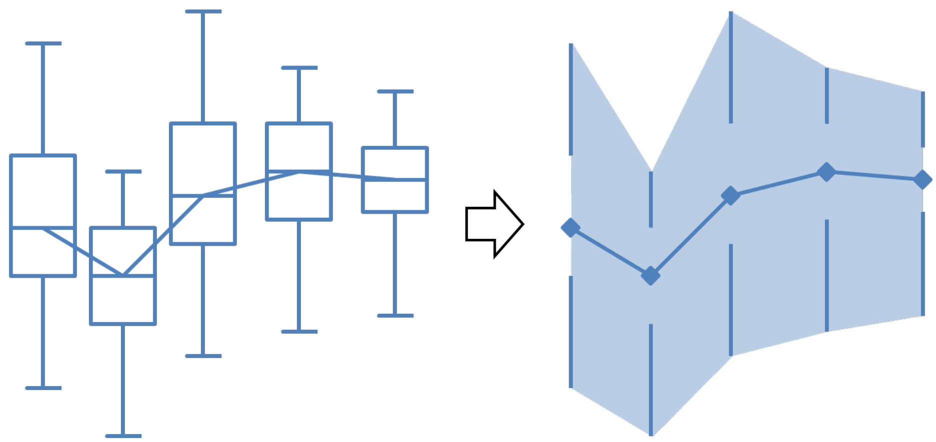


Figure 66: Simplified Box Plots Explanation

design was compared to the probability of recovery seen in the truth model. The differences between the recovery from the truth model and the recovery estimate for each design was calculated and designated the “Recovery Estimate Error.” To determine an aggregate measure of the number of random search mitigation cases required, statistical measures of this recovery estimate error were generated based on the entire set of designs. Figure 67 plots the results of this set of runs. The x-axis shows the number of random search mitigation points plotted on a logarithmic scale. For each run, a distribution was created and plotted using a simplified box plot described in Figure 66. The blue diamonds represent the median of each distribution. A line is drawn connecting these median values to show the trend of the estimate error with increasing number of cases. The blue vertical lines connect the inner quantiles of the distribution and the edges of each distribution, excluding outliers. A blue region higlights the change in the width of the distributions. The red line shows the run time required to evaluate 1,000 designs, with the time plotted (in hours) on the secondary y-axis.

Examination of this data shows that increasing the number of random search cases continuously decreases the recovery estimate error. However, this decrease in error comes at a cost of increasing run time required. The ARMOUR method needs an

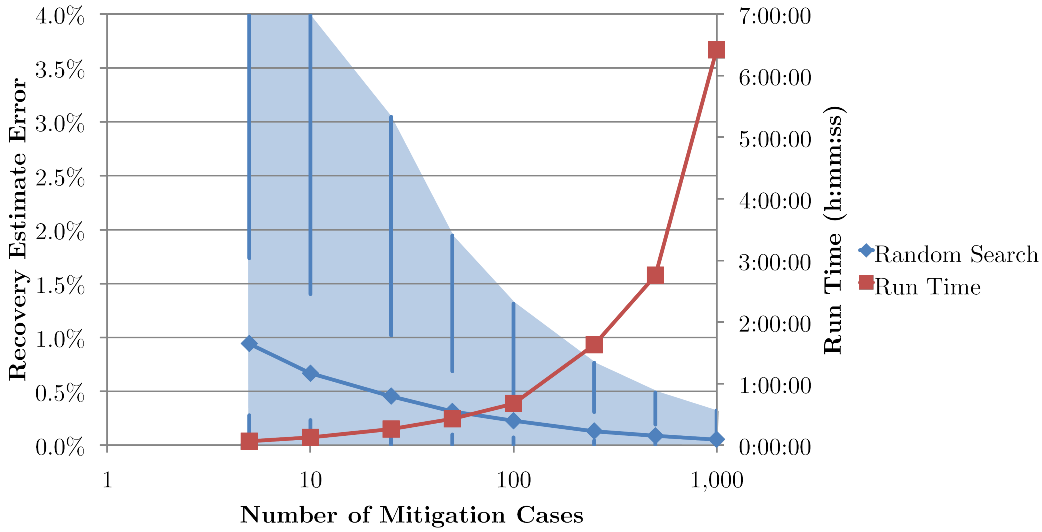


Figure 67: Recovery Estimate Error vs. Number of Random Mitigation Search Cases

accurate estimate of the probability of recovery in order to help the designer select the best aircraft. However, the ARMOUR method uses an optimization algorithm to determine the best design; thus, it is safe to assume that many designs will need to be examined. Thus, an undesirable tradeoff exists between the accuracy of the recovery estimate and the time required to perform this assessment. With the performance of this method in mind, other possible techniques will be examined to explore the mitigaiton space in the hopes of finding a more suitiable tradeoff between accuracy and execution time.

7.4.2 Recovery via Single Mitigation Actions

It is conceivable that a company constructing an aircraft would be prefer to limit the design team to implementing only one mitigation action during preliminary design. It is feasible to evaluate how the design space would be effected by such a decision. This is, of course, a completely valid assessment method. The difference between this and allowing the MA setting to be any combination of options will largely come down to corporate policy.

In this case, the mitigation space will restricted to a series of one-dimensional searches. As such, fewer points should be necessary to estimate the probability of recovery. Additionally, since the mitigation space is restricted, it is expected that the recovery probability will be reduced for at least some of the design space.

A set of one dimensional sweeps will be executed to estimate the probability of recovery. Each one dimensional sweep will evaluate one dimension of the possible mitigation actions. As in Section 7.4.1, a series of tests will be executed, evaluating the resolution necessary to accurately estimate the probability of recovery. Table 13 shows the number of one-dimensional search cases used for each run of the method. The total number of cases executed is also included for comparison to the number of cases used in the random search implementation.

Table 13: Number of 1D Search Mitigation Cases to Execute

Mitigation Search Method		Run 1	2	3	4	5	6	7	8
Random	Cases	5	10	25	50	100	250	500	1000
1D Search	Per Dimension	2	5	10	20	50	100		
	Total Cases	9	18	33	63	153	303		

Figure 68 shows the recovery estimate error this one-dimensional search in red compared to the previous random search algorithm in blue. The x-axis shows the number of mitigation cases executed on a logarithmic scale. The data points show the median recovery estimate error for that number of mitigation cases for each method. For each run, a distribution was created and plotted using a simplified box plot described in Figure 66. The symbols represent the median of each distribution and are connected to show the trend of the median error. The vertical lines connect the inner quantiles of the distribution and the edges of each distribution, excluding outliers. The blue region higlights the distributions from the random search exploration. The red region designates the distribtions from the one-dimensional search method.

From Figure 68 it is evident that the recovery estimate of the one-dimensional mitigation assessment is much more accurate than a random search for similar numbers of cases when the number of mitigation cases is low. Further, the one-dimensional search

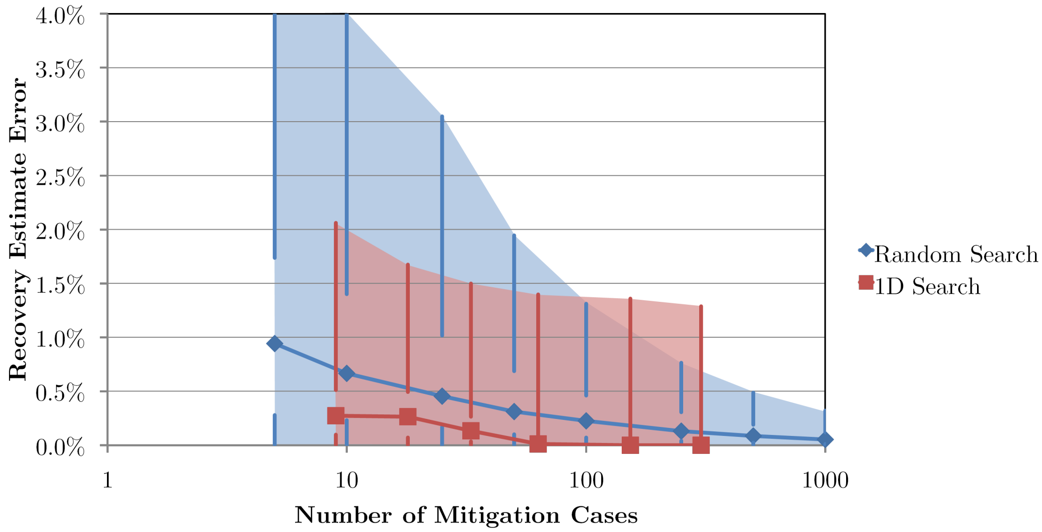


Figure 68: Recovery Estimate Error vs. Number of 1D Mitigation Cases

performs better for all percentiles except the extreme top end of the distributions. This is likely due to the fact that the single dimension mitigation search is guaranteed to explore its very constrained mitigation space evenly, while the random search method may not explore all regions equally. It is expected that the one-dimensional search method should perform worse when investigating designs which have multiple active constraints, which will be explored shortly.

Comparing the run time associated with the one-dimensional search to the random search data yields somewhat unsurprising results. Figure 69 compares these execution times on the y-axis versus the number of mitigation cases on the x-axis. The random search implementation is shown in blue while the one-dimensional search is shown in red. As the two lines lie on top of one another, the execution time depends primarily upon the number of mitigation cases, and is roughly the same for both implementations. The slight difference seen in execution time is believed to be a result of the user's actions in other programs at the time of execution.

From this data alone, an assumption could be made at this point that running a series of one-dimensional sweep of single mitigation actions may be more efficient and/or more accurate than running a random search throughout the mitigation space.

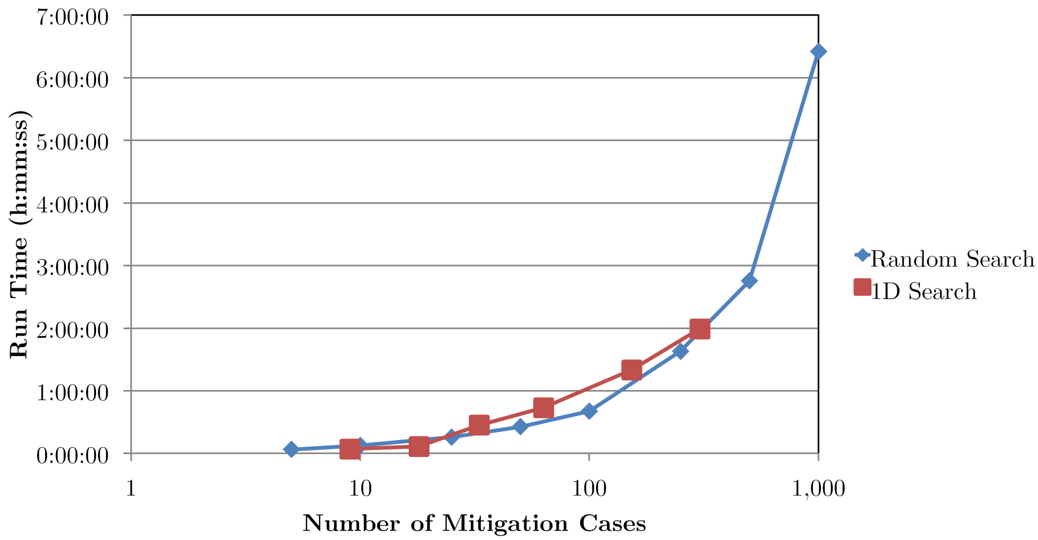


Figure 69: Run Time vs. Number of 1D Mitigation Cases

The evidence shows that this is generally true; however, assuming that this will always be true may be a significant misstep. Specifically, this method is expected to do poorly in any regions of the design space where multiple recoverable constraints are violated. In aircraft conceptual design, multiple constraints are frequently active or nearly active for the selected vehicle, so this multiple-constraint region is an area which is expected to be useful to a design study.

To examine this possibility, a subset of the previously executed cases were examined. The selected cases had a non-zero probability of compliance. Additionally, these cases had non-unity probabilities of compliance for the two most easily recoverable constraints: approach speed and rate of climb. Examining this small set of cases in the same manner as before yields Figure 70. The recovery estimates are compared to the recoveries seen in the "truth model" established earlier. This recovery estimate error is plotted on the y-axis versus the number of mitigation cases executed on the x-axis. Again, random search is plotted in blue while the one-dimensional mitigation search is plotted in red. The median recovery error for this set of cases is shown by the symbol at the middle of the vertical lines. The vertical lines themselves represent the distribution of this error when viewed across the design space.

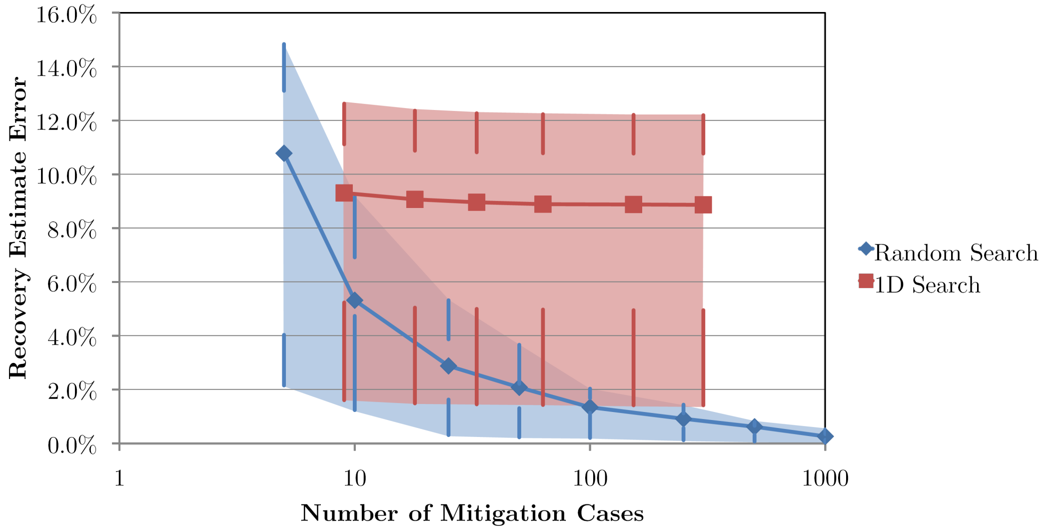


Figure 70: Recovery Estimate Error for Cases with Multiple Constraint Violations

As expected, it is clear from this figure that the random search algorithm has more trouble than in the rest of the design space, as evidenced by the magnitude of the y-axis; however, it does converge steadily with more cases. Conversely, the one-dimensional mitigation search method shows significant probability of recovery error. Furthermore, the estimate does not improve significantly with increased numbers of cases.

It appears that a one-dimensional mitigation search is both more efficient than a random search and yields similar results for the majority of the design space. Unfortunately, the method does fail to account for actual recoverable space when multiple constraints are active. Again, a one-dimensional search may be insightful in situations where the design team is uncomfortable implementing multiple mitigation actions simultaneously. In such a case, this option would be especially beneficial for both speed and pragmatic reasons.

It is also possible that a method could be devised which switches between one-dimensional and random searches as required by the design space. This concept is appealing for time-saving reasons and could be worth future consideration.

7.4.3 Recovery via Hybrid Mitigation Method

The random search appears very good at recovering difficult designs. Conversely, 1D search appears to be very quick and performs more consistently with fewer numbers of mitigation cases. Unfortunately, the one-dimensional search algorithm is very bad at recovering difficult designs. In hope of obtaining the advantages of both methods simultaneously, a hybrid method was posed. This method uses the one-dimensional search algorithm augmented by a scattering of random search points. Table 14 shows the number of cases executed for each run of the hybrid methodology, including a breakdown of the number of one-dimensional search cases and the number of random search cases for each run.

Table 14: Number of Hybrid Search Mitigation Cases to Execute

Mitigation Search Method		Run 1	2	3	4	5	6	7	8
Random	Cases	5	10	25	50	100	250	500	1000
1D Search	Per Dimension	2	5	10	20	50	100		
	Total Cases	9	18	33	63	153	303		
Hybrid	Per Dimension	2	5	10	20				
	Random Cases	20	50	100	200				
	Total Cases	26	65	130	260				

Figure 71 shows how the hybrid methodology performs against the other mitigation implementations for the "difficult" cases discussed in Section 7.4.2. Again, the number of mitigation cases is plotted on the x-axis on a logarithmic scale. The recovery estimate error is shown on the y-axis. The random search and one-dimensional search results from Figure 68 are shown in blue and red, respectively. The green points and lines show the result of the new hybrid implementation. The symbols once again show the median error. The vertical lines show the distribution from quantile to extreme. Each set of distributions is contained within a region of corresponding color.

In general, the hybrid method shows the same expected recovery estimate error as the one-dimensional method. This is a good sign that the hybrid method shows

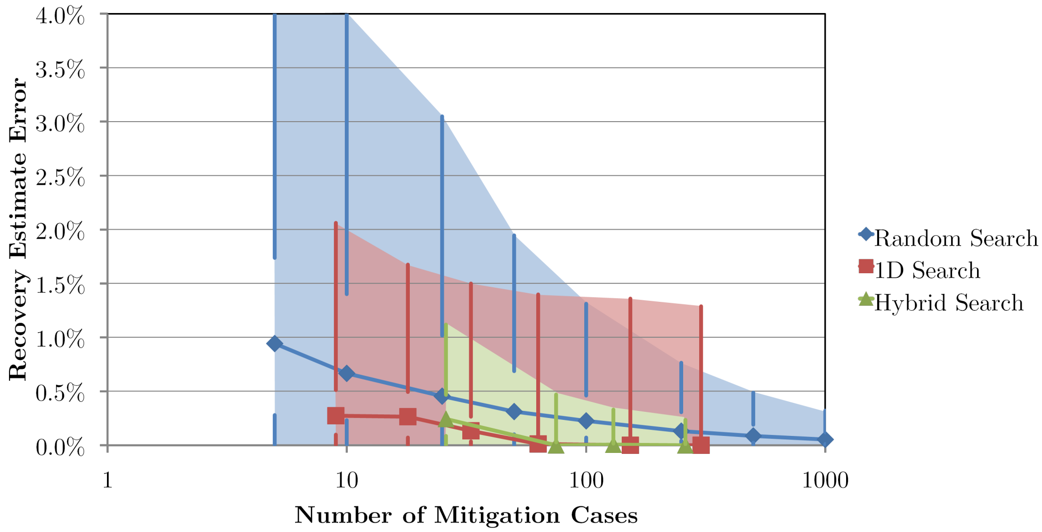


Figure 71: Recovery Estimate Error vs. Number of Hybrid (1D and Random) Mitigation Cases

some promise. The run time of the hybrid methodology should be explored. Further, it is necessary to investigate the Achilles' heel of the one-dimensional mitigation implementation: cases with multiple active constraints.

The time requirements for each of the non-optimization methods is plotted in Figure 72. The x-axis shows the total number of mitigation points assessed for a given method. The y-axis shows the required time needed to execute the set of designs in hours. The three lines show variation in the amount of time needed to assess the same number of mitigation cases; however, it is believed that this variability is due to the use of different machines to perform the calculations and not any inherent advantage of one method over the other.

Figure 73 shows how the hybrid methodology performs against the other mitigation implementations for the "difficult" cases discussed in Section 7.4.2. Again, the number of mitigation cases is plotted on the x-axis on a logarithmic scale. The recovery estimate error is shown on the y-axis. The random search and one-dimensional search results from Figure 70 are shown in blue and red, respectively. Green shows the result of the new hybrid implementation. The symbols once again show the median

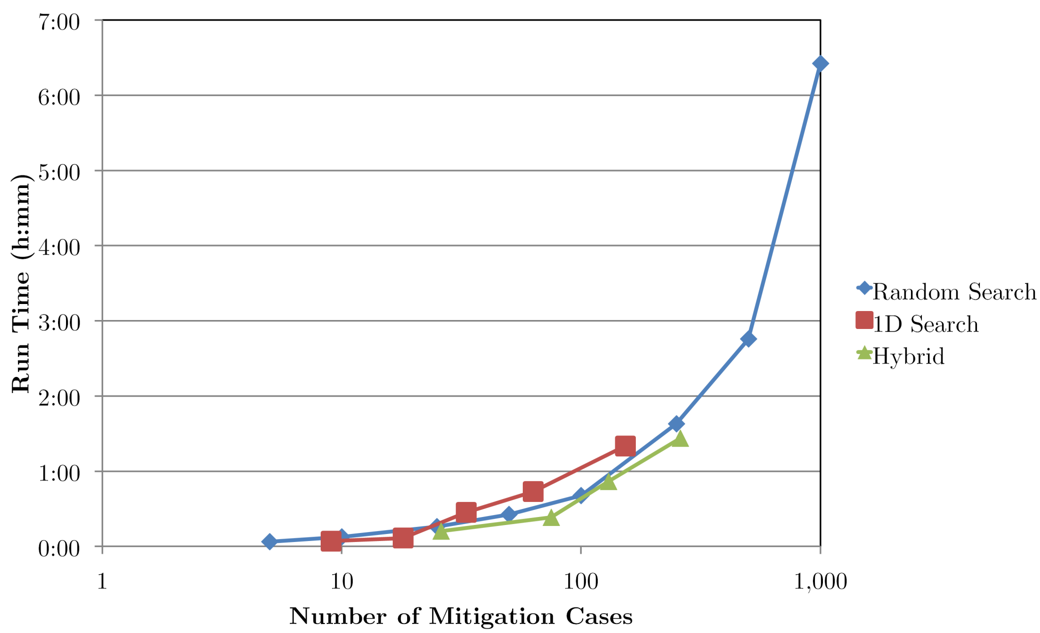


Figure 72: Run Time vs. Number of Hybrid (1D and Random) Mitigation Cases

error. The vertical lines illustrate the shape of each distribution and are contained in a colored region.

This hybridized method combining one-dimensional mitigation searches and a random search through the mitigation space appears to be superior to both methods. The hybrid method is at least as accurate as either of the other competing methods at estimating the probability of recovery for the general design space. Further, the hybrid method appears to perform better than either of the other methods when considering designs which violate multiple constraints. Finally, it accomplishes this level of performance without any notable penalty on the run time required to execute the analysis.

Further examination of the implementation is unhelpful at this stage. The example problem employed has a number of limitations: vastly different recovery capabilities for different constraint violations; few uncertain constraints; lack of industry data. Due to these limitations, any additional insight would likely be too problem-specific to be generalizable to other problems. Instead, this investigation has shown the

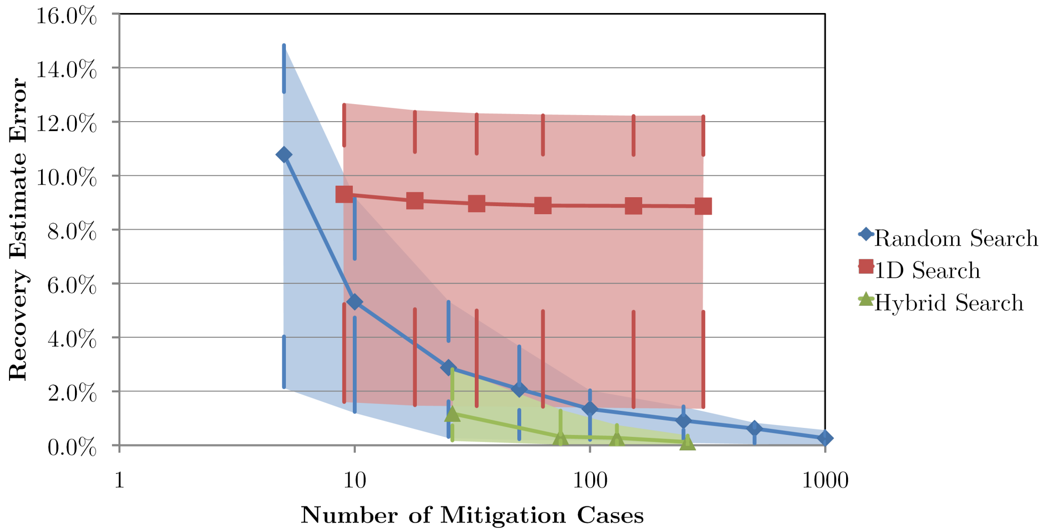


Figure 73: Recovery Estimate Error for Cases with Multiple Constraint Violations

promise of such a hybrid implementation (speed, accuracy, ability to deal with difficult portions of the design space). It is recommended that any future implementation of this methodology investigate the best hybrid approach for its own specific problem.

This hybrid one-dimensional and random search implementation will be used in the implementation developed in Chapter 6. This strategy affords the advantage that the recovery assessment (i.e. mitigation action analyses) will take a known, fixed amount of time.

7.5 Impact of Mitigation Penalty

The penalties imposed in Section 6.4 are assumed values. Thus, it would be beneficial to examine the extent of their impact on the probability of recovery seen by designs. To investigate this, the same design space exploration used in Section 7.4 was explored again.

A test of reducing the penalty associated with additional fuel was executed. The penalty imposed in Section 6.4.1 was reduced by a factor of four. This reduction lowered the amount of empty weight added to the aircraft for the same amount of fuel weight. Furthermore, it allowed the mitigation action further range of applicability before the additional weight on the aircraft was no longer fuel weight.

This mitigation action is intended to fix the range constraint violation. Thus, it is expected that reducing the associated penalty and increasing the range of applicability will increase the probability of recovery for any designs where range was an active constraint. The design space exploration from Section 7.4 was executed with this reduced penalty. After the data was generated, designs were grouped based on their most stringent compliance constraint – the constraint with which the design was least compliant compared to the other constraints.

The probability of recovery for a given design with the reduced penalty mentioned above was compared to the probability of recovery measure when the full mitigation penalty was imposed. By subtracting the new probability of recovery from the old and dividing by the old recovery, a multiplicative factor was constructed which shows how much more recovery was seen by reducing the mitigation penalty. The designs were then grouped based on whether or not range was the most stringent compliance constraint.

Figure 74 shows the results of this study. The two columns correspond to whether or not range was the most stringent active constraint for that design. The y-axis plots the additional recovery is seen by reducing the penalty applied to the fuel addition mitigation action. The bar chart shows the additional recovery, averaged over all designs and normalized by the design's recovery under normal penalty conditions. The error bars indicate the variability of this recovery when examining different designs by showing plus or minus one standard deviation.

It is clear that a reduction in mitigation penalty can have a very large impact on the probability of recovery. Reduction of the penalty does not change the basic behavior of the mitigation action. The post-sizing fuel mitigation action still recovers the same performance shortfall as with the normal penalty. Instead, it simply makes the mitigation action more effective by giving it a further range of applicability and

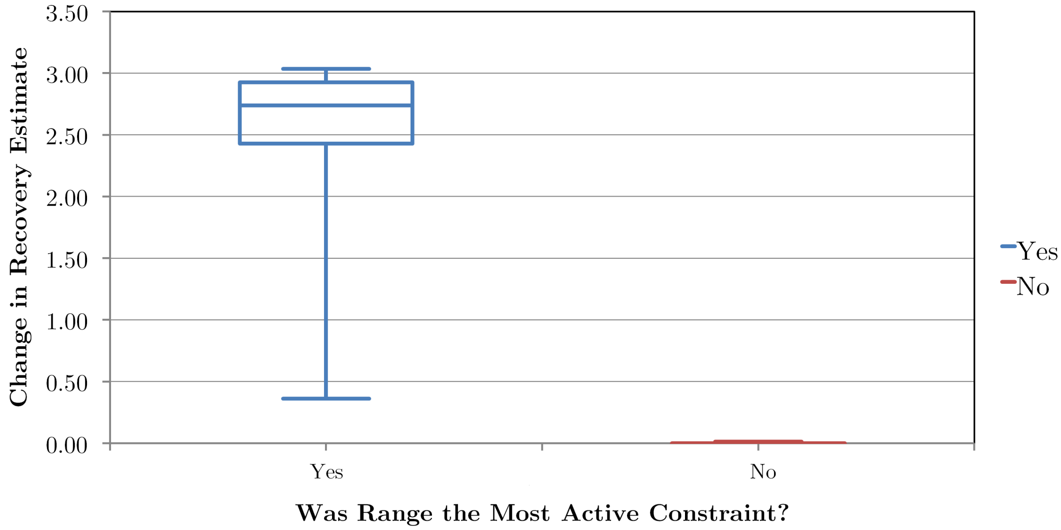


Figure 74: Change in Recovery Estimate with Reduced Mitigation Penalty

causing fewer conflicts between competing performance constraints.

7.6 Design Space Exploration

A large design space exploration will be performed to explore the trade space between expected block fuel, probability of compliance, and probability of recovery. To perform this exploration, a Monte Carlo simulation of the design space is constructed. Each of these resulting design points is executed through the framework discussed in Chapter 6. The reliability of each point will be assessed through a different Monte Carlo Simulation on the uncertainty space. The failed uncertainty scenarios are evaluated through the mitigation space exploration to see how many can be recovered. This information is compiled as discussed in Section 3.5 to calculate the expected economic block fuel and the probability of success. It is expected that multiple designs which have similar probabilities of compliance and expected block fuel will have different probabilities of recovery. This will occur when the designs have different active constraints.

Figure 75 shows the results of an exploration of the design space including all design and mitigation variables. These designs are assessed through the entire process described in Chapter 6. The resulting expected block fuel and reliabilities are plotted on this chart. The x-axis shows the expected economic range block fuel for a given design when measured across all uncertainty scenarios. On the y-axis multiple probabilities are displayed simultaneously. The probability of compliance for the design, the probability that the design will meet all constraints without any mitigation actions, is displayed as a blue diamond. For the same design, a vertical green line indicates the additional probability of recovery possible through the considered mitigation actions, which must by definition be strictly positive. The triangle at the top of this green line represents the total probability of success of the design, i.e. the probability that the design can meet all constraints either without or through the application of mitigation actions. Designs with less than 75 percent probability of compliance or with greater than an expected 114,000 pounds of economic range block fuel were excluded from the plot.

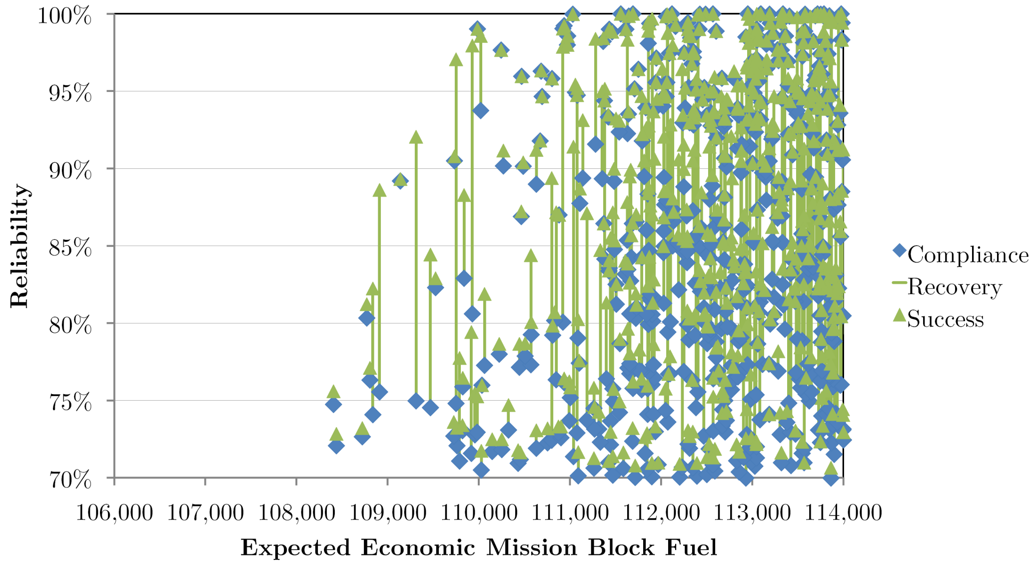


Figure 75: Design Space Exploration with Recovery

Figure 76 shows the same diagram with the details annotated. A specific design is highlighted. The probability of compliance is pointed out at the bottom of the design. The probability of success shown at the top. The length of the line between these data points corresponds to the probability of recovery for that same design.

Now that the design space has been viewed in mass, other details can be investigated. It was expected that the probability of recovery of a design through mitigation actions would not necessarily correlate with the expected performance of that design or its probability of compliance. To examine whether or not a trend exists, it will be helpful to examine designs with similar probabilities of compliance. Figure 77 shows the same design space exploration as Figure 75; however, only designs with a similar level of compliance are included. Specifically, any designs with less than 75 percent probability of compliance or greater than 77 percent compliance were removed from the chart. The remaining designs still show their probability of compliance via a blue diamond. The green line extending from the design shows its probability of recovery. The green triangle at the top of the line shows the resulting probability of success for

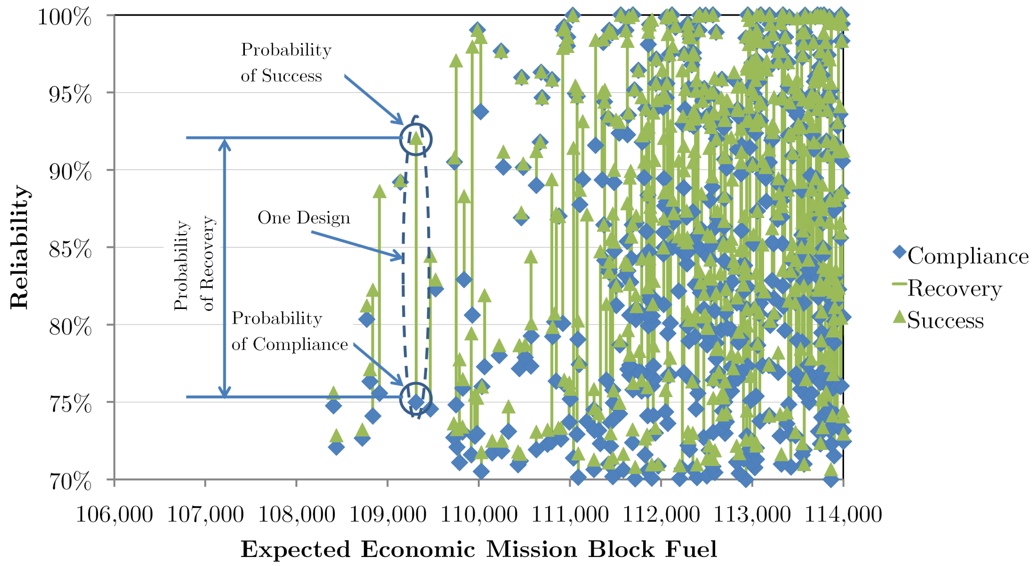


Figure 76: Design Space Exploration with Recovery - Annotated

that given design.

From Figure 77, it is possible to examine designs which will appear almost identical in a traditional reliability study and observe hidden differences between them. Examining the the left most two designs – the ones with the lowest expected economic block fuel – it appears that the designs would be almost indistinguishable via traditional RBDO. Indeed, the two designs differ by 110 lbs of expected block fuel and less than eight tenths of a percent probability of compliance. However, one design has less than one (1) percent probability of recovery through mitigation actions while the other has over 13 percent probability of recovery via those same mitigation actions. In the end, selecting one of these two "very similar" designs would yield an aircraft with either an 88 percent total probability of success including mitigation actions or one with on 77 percent probability of success. Again, this level of information is *completely new* for a conceptual design study and could be very useful when deciding which design to move forward with for development.

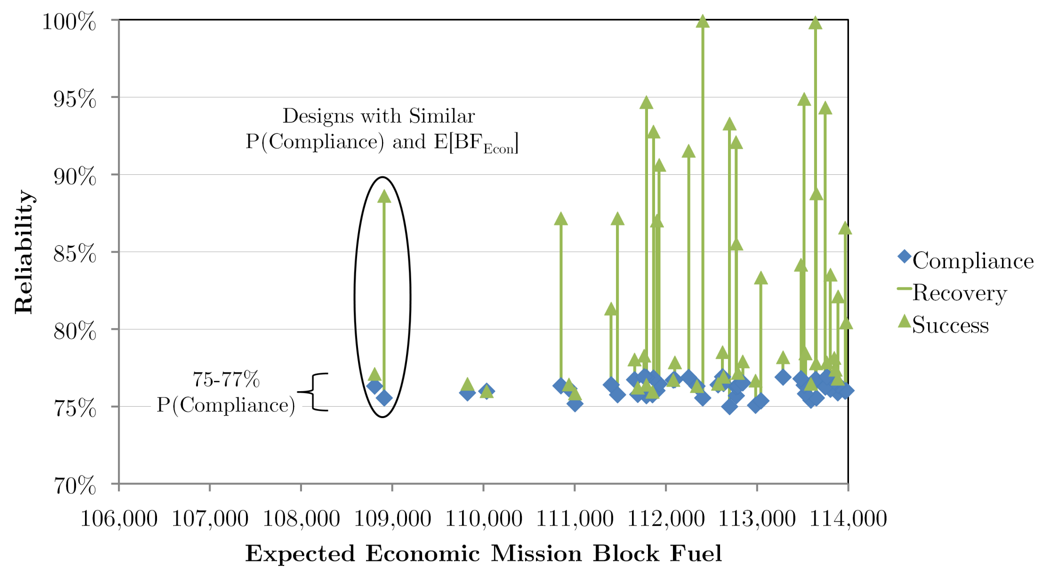


Figure 77: Design Space Exploration - 75% to 77% Compliance

7.7 Pareto Frontiers

It is evident from Figure 75 that a trade-off exists between reliability goals and the expected block fuel for this design space. Such behavior is expected and implies that a single objective optimization in one of these dimensions will increasingly penalize the other objective. This concept is reminiscent of the Pareto Frontier discussed in Section 2.5. Thus, it may be beneficial for a designer to investigate in some sort of Pareto Frontier finding algorithm to find the "best" designs from which to select a final concept.

Ideally, the designer would have the ability to select any probability of compliance, probability of success, and expected block fuel that he or she wanted. Anyone familiar with aircraft design, or any kind of design for that matter, knows that this is an unreasonable expectation. Instead, it is expected that there will necessarily be a trade-off between competing objectives and that some hard decisions will need to be made. To make this decision, the trade-off between these competing objectives must be quantified in some manner. This trade-off is the same trade-off as the one discussed in Section 2.5 – a Pareto Frontier.

To find a Pareto Frontier for the process in question, an optimizer was attached to the implementation described in Chapter 6 to find an optimal set of designs. This optimizer employed a genetic algorithm to examine the design space to find the best possible set of designs given an objective function and constraints on the result. The optimizer was given access to both design variables and margin, allowing it full control over the design selected. For any selected design, the implementation returned the expected economic range block fuel along with probabilities of compliance and success.

In order to construct a Pareto Frontier, a sweep through the objective space was performed. A single objective was selected: expected block fuel. Each time the optimizer was called, it would try to minimize this objective. To perform the sweep and create a Pareto Frontier, a constraint was imposed on one of the reliability measures. Each time the optimizer was called, a different minimum level of reliability was required, constraining the design space available to the optimizer.

The Pareto Frontier shown in Figure 78 was constructed using an optimizer looking for the minimum expected block fuel ($\mathbb{E}[BF_{Econ}(x, h)]$), shown on the x-axis. For each call to the optimizer, a minimum level of probability of compliance was imposed. This minimum level of compliance was different for each optimization and ranged from 75 percent compliance up to 100 percent compliance, in increments of one percent. It was not mandated that the optimizer meet this minimum level of compliance; the optimizer merely needed to exceed it. The resulting reliabilities for the final selected designs is shown on the y-axis. Additionally, the probabilities of recovery and success were determined for each resulting design. For a given design, the lower point corresponds to the probability of compliance, while the higher point corresponds to the probability of success. The line connecting these two points is the probability of recovery for that design. The results of the design space exploration from Section 7.6 are also shown on this chart for reference. They are colored entirely in green, but still show the probabilities of compliance, recovery, and success for each design.

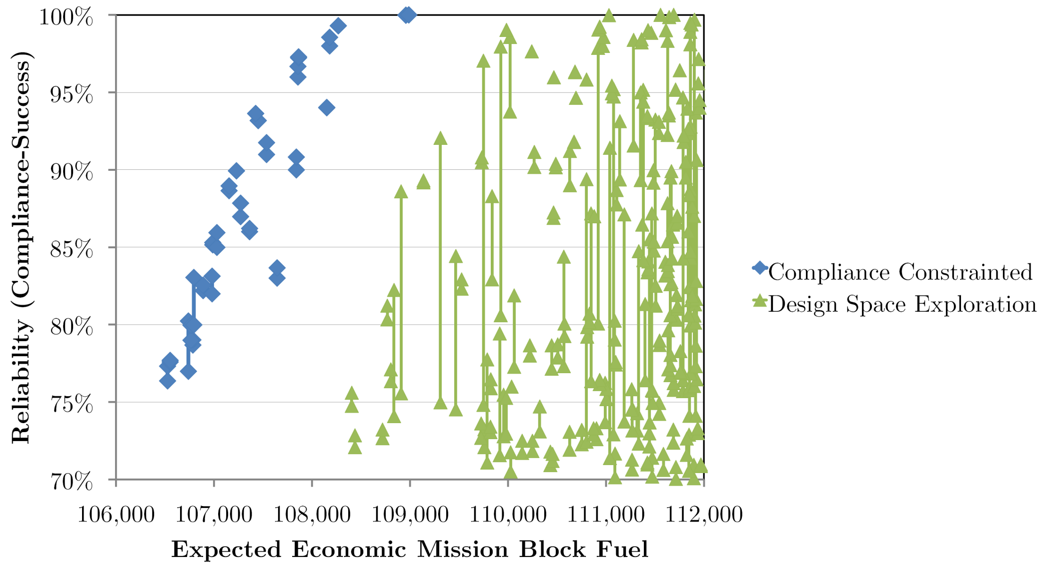


Figure 78: Probability of Compliance vs. Expected Block Fuel Trade Space

Using the capabilities enabled by the proposed methodology, a Pareto frontier was constructed between the expected block fuel objective function and the overall probability of success, accounting for mitigation actions. To construct this frontier an optimizer operated on the design and margin variables, searching for the lowest expected block fuel while maintaining at least a minimum level of probability of success and probability of compliance. For all optimizer runs, the optimizer was constrained by a minimum of 70 percent probability of compliance. Each optimization was given a different minimum level of probability of success, ranging from 75 percent up to 100 percent in one percent increments.

This new Pareto frontier between expected block fuel and probability of compliance is shown in Figure 79 in red. The reliabilities for a design are once again shown on the y-axis versus the expected block fuel for the design on the x-axis. For each individual design the probability of compliance is shown as the lower red block, the probability of success is shown as the higher red block, and the probability of recovery is the red line connecting the two points. The Pareto frontier established for Figure 78 and the design space exploration from Section 7.6 are shown for comparison.

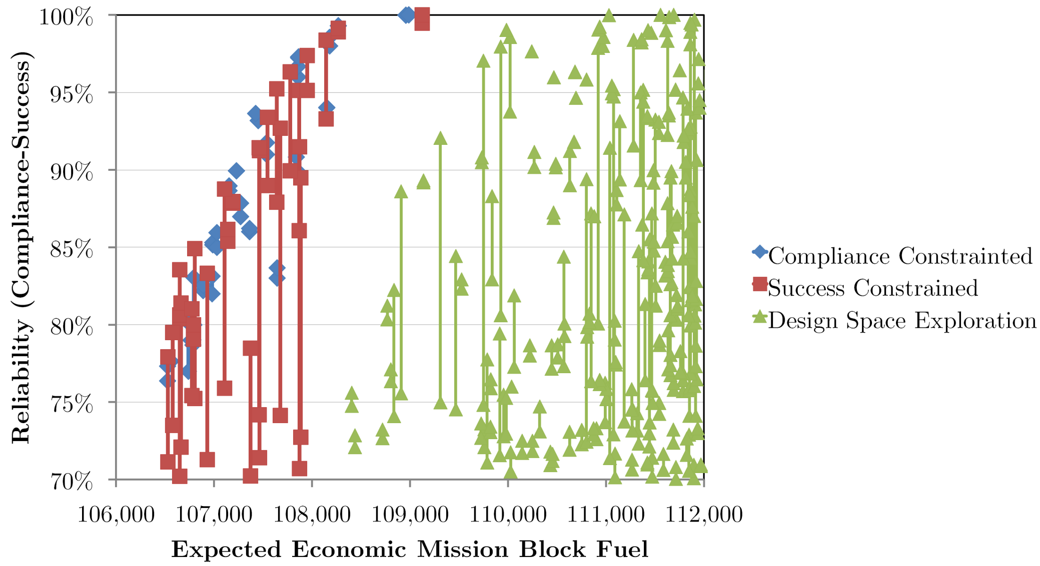


Figure 79: Probability of Success vs. Expected Block Fuel Trade Space

7.8 Chapter Summary

This chapter containts a demonstration of some of the capabilities enabled by the ARMOUR methodology. Exercising the formulation from Chapter 5 for a specific example of the conceptual design of a large civil transport aircraft developed in Chapter 6 allows for multiple trade studies and assessments which could not have been performed previously. By including uncertainty margin in an uncertainty quantification framework, the impact of uncertainty margins on the probability of compliance could be investigated. The addition of mitigation actions allowed for the assessment of the impact that uncertainty margins have on the probability of recovery across the design space. The importance of accurately estimating appropriate penalty functions on these mitigation actions was shown by studying the impact of changing a mitigation penalty. The design space was explored using the new methodology, allowing for previously indistinguishable designs to be separated by the new metric of their probability of recovery. This new methodology was used to select designs optimized using the new metrics, which allows for the visualization of a trade space which has always existed but has never been seen before now.

CHAPTER VIII

CONCLUSIONS

This chapter examines the contributions of this thesis to the state of the art in aircraft design. The research questions and hypotheses developed in Chapter 3 are reviewed and the evidence presented in Chapter 4 to support or refute each of them is recalled. Finally, future opportunities arising from the concepts explored in this work are discussed.

8.1 Review of Research Questions and Hypotheses

The motivating problem for this work was defined in Section 1.4. In brief, it states that uncertainty during the aircraft design process can lead to the selection of a vehicle which will to fail to meet its performance constraints when analyzed in detail during later design stages, even with preventative measures like uncertainty margins in place. This occurs because the conceptual design is typically executed with faster, less detailed, and deterministic design tools. Chapter 2 illustrated that uncertainty quantification methods like RBDO have the potential to address some of this design uncertainty.

However, existing methods do not explicitly take into account some of the realities of aircraft design. A sophisticated sizing process, frequently a central tenant of conceptual design is not accounted for by the processes. There is limited treatment existing of uncertainty margins and traditional methods which do treat these margins do not allow for the quantitative assessment of the resulting reliability. Most existing RBDO methods do not allow for the assessment of the impact of mitigation actions; those methods that do account for late stage design changes do so either at the expense of either abolishing the design freeze present in the stages of aircraft design or with significant additional costs in terms of detailed analyses which are difficult to perform during conceptual design. Thus, in Section 2.7 the following Research Objective was developed.

> **Research Objective:** Quantify uncertainty during the design process for a new aircraft, including a sophisticated sizing analysis, uncertainty margins, and mitigation actions.

In order to develop a methodology to achieve this objective, the problem was broken down into smaller components. Specific focus was given to design uncertainty given the stages of aircraft design, the treatment of uncertainty margins in a probabilistic framework, and the concept of mitigation actions – late stage-design, non-ideal changes to a design made to bring the aircraft back into compliance with constraints.

8.1.1 Treatment of Design Uncertainty with Aircraft Sizing

Upon comparing the stages of aircraft design to the available uncertainty quantification literature, it became clear that simply modeling the uncertainty in a rigorous way while paying heed to the requirements of the design process was in itself a non-trivial problem, and that this concept would need to be formalized as in Research Question 1.

> **Research Question 1** *How should aircraft design with uncertainty be modeled for reliability analysis, accounting for the stages of design?*

Detailed examination of the stages of aircraft design and the implied outcomes of modeling uncertainty during processes representing various phases was discussed. The implications of applying an uncertainty variable to either sizing or performance analysis were explored and, ultimately, the concept of uncertainty during the stages of aircraft design aligned with sizing the vehicle, freezing its configuration and design parameters, and only then doing a performance analysis as stated in Hypothesis 1.

Hypothesis 1 *To emulate the aircraft conceptual and preliminary design process, uncertainty must be implemented after the sizing of the aircraft is complete. Modeling uncertainty during sizing will yield incorrect results for aircraft conceptual design under uncertainty.*

The experiment in Section 4.1 tested this hypothesis. A simple weight-based canonical sizing problem was developed based on equations from aircraft design literature. The problem was augmented with two different uncertainty implementations: modeling uncertainty during sizing and during performance analysis. These two implementations were compared, and the results showed that the two implementations differed greatly in their effect on resulting aircraft weights. Further, by addressing the implications of these weights on different performance parameters, the difference in vehicle performance between these two implementations was clarified. In the end, modeling during sizing was determined to have significant deficiencies, leading to the result that uncertainty quantification must be measured by implementing uncertainty scenarios during the performance analysis, after sizing is completed, justifying Hypothesis 1.

8.1.2 Margins as Design Variables in a Probabilistic Formulation

The traditional use of uncertainty margins in a deterministic design process as a method to account for uncertainty was examined. Further, it was determined that uncertainty margins had uses during later design stages and, thus, it would be beneficial to include these margins during a reliability analysis. It was hoped, as inquired in Research Question 2, that this margin could be integrated into a reliability study and that such a study could actually be used to *set* uncertainty margins.

Research Question 2 *Is it possible to select a desired probability of compliance and then quantitatively determine a level of margin which will yield that probability of compliance?*

To answer Research Question 2, uncertainty quantification studies were examined which explored concepts similar to uncertainty margins (in the form of safety factors) by assessing overall reliability. It was therefore theorized in Hypothesis 2 that margins can be treated similarly and that reliability goals could be used to set uncertainty margins.

> **Hypothesis 2** *By including an uncertainty margin during the sizing process and removing it (but not its effect) before the uncertainty analysis, the impact of margin on the probability of compliance can be seen. Using an optimizer, it will be straight-forward to determine an appropriate level of margin to achieve a desired probability of compliance.*

Experiment 2 was designed to test the validity of Hypothesis 2. A partially constructed version of the implementation from Chapter 6 was used to test the ability to use the established reliability framework to set uncertainty margins. It was found that if a level of margin exists which can satisfy a probability of compliance goal, an optimizer is easily able to find that level of margin. Further, by allowing the optimizer to adjust both design variables and uncertainty margins simultaneously, the optimizer will produce even better designs with higher reliability, better performance, or both.

8.1.3 Quantification of Probability of Recovery

This work was motivated by the concept of mitigation actions. An important overall goal was to introduce these mitigation actions into a probabilistic aircraft conceptual design process. Very few studies with similar goals could be located, and they did not account for actions after a design freeze. Since no suitable framework existed, a new one must be developed, motivating Research Question 3.

> **Research Question 3** *How should mitigation be represented in a probabilistic conceptual design model?*

Based on the thought experiment in Section 3.3.2, Hypothesis 3 was developed.

> **Hypothesis 3** *It will be necessary to perform a mitigation assessment for each failed uncertainty outcome to get an accurate determination of the probability of recovery.*

This hypothesis was tested in Section 4.2. A canonical problem was developed using equations from aircraft design texts. This canonical problem was intended to assess the uncertainty space and resulting mitigation spaces of a single design. Using these spaces, different uncertainty scenarios were implemented and their resulting mitigation space investigated. It was shown that even with only one mitigation action, the required level of mitigation action needed would be different depending on the uncertainty scenario investigated. This was further evaluated to show that the maximum recovery for an "overall best" mitigation action setting would provide lower recovery than investigating mitigation actions separately for all designs, supporting Hypothesis 3.

Finally, Research Question 4 was asked to inform the needs of the specific implementation of a reliability analysis that includes mitigation actions.

> **Research Question 4** *What reliability assessment method should be used to model the aircraft Conceptual and preliminary design process with mitigation actions?*

The support for Hypothesis 3 meant that the reliability assessment method must allow for the sampling of individual uncertainty scenarios. Thus, a sampling method must be implemented to assess the reliability of a design, and other methods like boundary approximation methods did not appear to be feasible for this method. Of the sampling methods, Monte Carlo simulation was selected due to its relative ease of implementation and the ability to emulate any desired distribution on uncertainty variables.

Based on the results of Hypothesis 1 to 3, a new methodology was developed to fix the identified gaps. This new Aircraft Recovery through Mitigation & Optimization under Uncertainty for Reliability (ARMOUR) method incorporated aircraft sizing, uncertainty margins, and mitigation actions into a reliability framework. This new method gives the decision maker more information than could previously be gleaned from preliminary design. Reliability methods are an initial and integral step to understanding the results of the uncertainty experienced in the conceptual design process; however, they do not account for actions taken by engineers during later stages of the process which could remedy encountered problems – mitigation actions. These actions, which have not been included in reliability analyses previously, allow for the calculation of a probability of recovery from encountered failures. This probability of recovery can be combined with the probability of compliance through traditional RBDO to determine the overall probability of success for an aircraft design.

With this new information, a decision maker gains information about how likely a design is to be successful in complying with all performance requirements, including preliminary design mitigation actions, as well as how likely it is that a design would be compliant with performance requirements without these reactions. These probabilities can be traded with traditional aircraft design objective functions to allow the designer to select the combination of performance and reliability which which s/he is most comfortable. In the end, the decision maker can select designs which perform better against traditional metrics, designs which have a lower probability of failure, or some combination of the two.

The ARMOUR method was implemented for the concpetual design of a large civil transport aircraft. Design variables, margins, uncertainty variables, mitigation actions, and mitigation penalties were established. Performance responses, a design objective, and reliability requirements were established for the aircraft. A physics-based aircraft model was developed which used a set of aircraft and engine design tools to emulate the design process. The resulting data were regressed against the set of inputs to generate surrogate models to improve the speed of the analyses. These surrogates were integrated into a newly constructed uncertainty quantification and management framework which allowed for the calculation of the performance and reliability of any design as well as the optimization of the designs selected.

Substantiating Hypothesis 1 to 3 and extracting new, useful information from the resulting implementation demonstrates that this work has successfully addressed the stated research objective. The impacts of mitigation actions are assessed probabilistically in a conceptual design study by using a reliability framework. The inclusion of a mitigation exploration emulates the process by which a design team would attempt to recover aircraft which fail to meet their requirements. By evaluating these situations which could occur in preliminary design and bringing their impacts into conceptual design, the decision maker can now account for the probability of recovery in addition to expected performance and probability of compliance when selecting an initial design to develop. The inclusion of mitigation actions in a reliability framework will allow the decision maker to know with confidence whether a design can be fixed with the selected mitigation actions. This new information, which can be made available during conceptual design, brings knowledge forward in the design process and fulfils the underlying motivation of this thesis.

8.2 Contributions to the State of the Art

The Aircraft Recovery through Mitigation & Optimization under Uncertainty for Reliability (ARMOUR) method has been developed to assess the reliability of an aircraft under uncertainty during conceptual design. The ARMOUR method accounts for the use of a sophisticated aircraft sizing process. The method indicates the order in which design variables and uncertainty variables must be applied with regards to sizing and performance analysis. The potential error associated with incorrectly implementing these steps is quantified.

ARMOUR incorporates the ability to set uncertainty margins into the aircraft reliability analysis. An implementation of uncertainty margins in the stages of reliability analysis is established which allows margins to treat all uncertainty variables rather than directly treating only uncertainty variables which affect the same direct parameters as the margin. This allows for uncertainty margins to be set to specify a desired probability of compliance of a given design. By integrating this information into an optimizer, design variables can be varied simultaneously with uncertainty margins to select designs which have the best combination of performance and reliability.

By comparing real aircraft and engine design processes to reliability analyses in the literature, it was identified that existing methods have no tools which can account for actions taken by the preliminary design team to recover failed designs. These actions were dubbed "mitigation actions" and a method was proposed to incorporate these mitigation actions into an uncertain aircraft design process. The ARMOUR method allows for the calculation of a probability of recovery through these mitigation actions.

The construction of the ARMOUR methodology lead to a new reliability metric: the probability of recovery ($P\left(Recovery \mid x, h\right)$). This metric indicates the likelihood that mitigation actions will be bring the design back into compliance with performance constraints when the inherent uncertainty in the design process has cause the aircraft to violate one or more of these constraints. Combining this new information with the probability of compliance ($P\left(Compliance \mid x, h\right)$) generated through standard reliability methods allows for the calculation of the total probability of success ($P\left(Success \mid x, h\right)$) of a given design. The probability of success indicates the likelihood that an aircraft will meet all performance constraints. This metric accounts for both design uncertainty and the ability of a preliminary design team to recover the performance of the design should a constraint be violated.

The ARMOUR methodology is formulated to allow for the optimization of a design with respect to traditional design metrics. Reliability goals in the form of probability of compliance and probability of success are enforced. This will enable the designer to limit the amount of risk to an acceptable level.

The ARMOUR methodology has been implemented for a large (300 passenger class) civil transport design. The methodology allows for the selection of design variables and uncertainty margins which meet reliability goals on both the compliance and success of the aircraft and simultaneously optimize a traditional aircraft design performance metric. This implementation of the ARMOUR methodology was used to explore and quantify the relationships between probability of compliance ($P\left(Compliance \mid x, h\right)$) and uncertainty margins. Attempts were also made to quantify the relationship between the probability of success ($P\left(Success \mid x, h\right)$) and these margins.

8.3 Future Work

This study has opened multiple avenues for future work. Possibilities which seem particularly fruitful include the examination of recovery estimation while using boundary approximation methods and the application of the current methodology to later stages of design.

It may prove useful to examine the quantitative change in accuracy of the probability of recovery estimate when using boundary approximation methods. If the use of boundary approximation methods can be shown not to adversely affect the probability of recovery estimation, then these methods may allow for the relaxation of some assumed restrictions which led to the selected implementation in this thesis. By using these boundary approximation methods, it may be possible to avoid the use of sampling methods to estimate the reliability of a design; this could decrease the required number of function calls to a large degree. This reduction in function calls may even allow aircraft design tools to be called directly without the need for surrogate models. Such an implementation may require significant changes to the assumed forms of Equations (19) to (23)

The ARMOUR method may be adaptable to handle other design problems. Engine design is an area which is particularly of interest. Engine design has many of the same problems associated with aircraft design. The design of an engine takes place under the presence of uncertainty due to the use of lower fidelity modeling during the initial conceptual phase. Engine design often goes through a sizing process, though engines are sized to match thrust targets rather than a mission range. Higher fidelity tools are very expensive to operate and require extensive data about the engine before being operated, preventing designers from using these tools during conceptual design. Further, an engine design problem in which mitigation actions were employed was found as a clear example in the literature [43]. Thus, many of the same benefits would exist for using ARMOUR on the conceptual design of an engine.

The methodology established here may have further applications during later design stages. If this method were used for conceptual design, an easy-to-update implementation would exist. During later design stages when more information is known about a previously uncertainty variable, the assumed distributions of uncertainty variables could be updated. With these updated distributions, the method could be executed again to update the probability of compliance and probability of recovery estimations for that design. This update could help a design team to know which constraints will be problematic as the design changes. Further, knowledge of the likelihood of these constraint violations could help the design team prepare potential mitigation actions as it becomes more and more apparent that a particular action may be needed. This concept is particularly appealing because in later design stages, the aircraft will be frozen. Thus, only a single design needs to be investigated. This will remove the outside loop optimization, allowing for the analysis to take place much more quickly. Because of the resulting decrease in the number of function calls required to analyze the probabilities of compliance and success, more detailed analyses may be used to further refine the extracted information.

APPENDIX A

HYPOTHESIS 3 TEST EQUATIONS

To test this hypothesis, a simplified example problem will be constructed. Two uncertain variables will be implemented: an empty weight factor (U_W) and a drag factor (U_D). These should to allow for sufficient variation while keeping the problem simple. The empty weight factor is expected to affect both range and approach speed equations. The drag factor is expected to only impact the Breguet range equation. Impacts from the drag factor on approach speed are expected to be negligible.

Only a single mitigation action will be considered: a post-sizing fuel increase with an associated penalty. The mitigation action is intended to improve range; however, the penalty will eventually restrict its effectiveness. It is expected that mitigation will degrade approach speed. The post-sizing fuel increase will directly affect the aircraft's maximum takeoff weight (MTOW). Also, an associated penalty will be imposed on the vehicle empty weight to account for the added structure necessary to carry the extra load. This penalty is expected to increase faster as the additional fuel weight increases.

A.1 Constraints

In this example, two constraints will be considered: the aircraft range at a design payload and the landing approach speed (Vapp). Range will be assessed using the Breguet range equation [75]. The approach speed with be estimated using the aircraft stall speed and appropriate scalar. These equations will form the backbone of the current analysis.

A.1.1 Approach speed

For the purposes of regulation, the approach speed (Vapp) of a vehicle, the speed just before landing, is calculated as 1.3 times the stall speed of the airplane[75]. The stall speed can be calculated as show in Equation (33).

$$Vapp = 1.3\sqrt{\frac{2}{\rho}\frac{W_{landing}}{S}\frac{1}{C_{L_{max}}}} \tag{33}$$

Stall speed is a function of the landing weight of the vehicle, its wing area, the local air density, and maximum lift coefficient of the vehicle. For this example, it is assumed that the local air density (ρ) is fixed. Additionally, the wing area (S) and maximum lift coefficient ($C_{L_{max}}$) will have been set during conceptual design and thus will not change in this example. Only the landing weight ($W_{landing}$) will be affected during this experiment.

A.1.2 Range

The Breguet range equation demonstrated in Equation (35) calculates the distance an aircraft can fly during its cruise segment.

$$R = \frac{V}{C_t}\left(\frac{L}{D}\right)\ln\left(\frac{W_{inital}}{W_{final}}\right) \tag{35}$$

The equation assumes that the aircraft will fly at a fixed speed (V) and lift-to-drag ratio $\left(\frac{L}{D}\right)$, optimizing altitude as the weight of the vehicle changes. The fuel consumption (C_t) of the engine is assumed to be constant throughout the cruise segment. The equation is also dependent on the ratio of the initial cruise weight to the final cruise weight $\left(\frac{W_{inital}}{W_{final}}\right)$, indicating how much fuel is consumed during flight.

A.2 Implementing Uncertainty and Mitigation

The maximum takeoff weight ($MTOW$) of a vehicle can be broken down into components of empty vehicle weight (W_E), payload weight (W_{PL}), crew weight (W_{Crew}), and fuel weight (W_{Fuel}).

$$W_0 = W_E + W_{PL} + W_{Crew} + W_{Fuel} \tag{54}$$

The payload weight will be set by the number of passengers, which is dictated by market forces and is assumed to remain constant. The crew weight is derived based on the number of crew required for the operation of the vehicle and flight attendants to take care of the passengers. The uncertainty in the weight of the passengers or crew will not be assessed in this study.

In order to assess the performance at different stages of the flight, fuel weight can broken down by flight segment. Generally, this breakdown will depend on the data available and the information to be derived.

$$\begin{aligned} W_{Fuel} = W_{Fuel_{TaxiOut}} + W_{Fuel_{Takeoff}} + W_{Fuel_{Climb}} + W_{Fuel_{Cruise}} \\ + W_{Fuel_{Descent}} + W_{Fuel_{Landing}} + W_{Fuel_{TaxiIn}} + W_{Fuel_{Reserve}} \end{aligned} \tag{55}$$

For the purposes of this example, fuel will be broken down as show in Equation (56) into fuel consumed before cruise, cruise fuel, descent fuel, and fuel after descent. Fuel consumed before cruise includes taxi out, takeoff, and climb. Fuel consumed after descent includes landing, taxi in, and reserves.

$$W_{Fuel} = W_{Fuel_{BeforeCruise}} + W_{Fuel_{Cruise}} + W_{Fuel_{Descent}} + W_{Fuel_{AfterDescent}} \tag{56}$$

This equation contains all the different segments required to assess both the vehicle range and approach speed.

The initial cruise weight can be found by subtracting the fuel used in pre-cruise flight segments from the maximum takeoff weight of the vehicle.

$$W_{initial} = W_0 - W_{Fuel_{BeforeCruise}} \tag{57}$$

The weight of the vehicle at the end of cruise is equal to the initial cruise weight minus the fuel used in cruise.

$$W_{final} = W_{initial} - W_{Fuel_{Cruise}} \tag{58}$$

The landing weight is simply the final cruise weight minus fuel used in descent.

$$W_{landing} = W_{final} - W_{Fuel_{Descent}} \tag{59}$$

Substituting from definitions of MTOW and fuel weight, landing weight can also be found by summing the empty weight, payload, crew, and post-touchdown fuel consumed, including unusable fuel.

$$W_{landing} = W_E + W_{PL} + W_{Crew} + W_{Fuel_{AfterDescent}} \tag{60}$$

The final cruise weight can be calculated as the sum the empty weight, payload, crew, and fuel consumed after cruise as shown in Equation (61).

$$W_{final} = W_E + W_{PL} + W_{Crew} + W_{Fuel_{Descent}} + W_{Fuel_{AfterDescent}} \tag{61}$$

$$W_{final} = W_{landing} + W_{F_{Descent}} \propto W_{ZFW} \tag{62}$$

The initial cruise weight is equal to the takeoff weight minus the fuel used before cruise.

$$\begin{aligned} W_{initial} &= W_0 - W_{Fuel_{BeforeCruise}} \\ &= W_E + W_{PL} + W_{Crew} + W_{Fuel} - W_{Fuel_{BeforeCruise}} \end{aligned} \tag{63}$$

$$W_{initial} \propto W_0 \tag{64}$$

A.2.1 Weight

Uncertainty associated with the empty weight will, logically, directly affect the empty weight of the vehicle. Taking U_{EW} to be the uncertain scenario associated with empty weight, the change in empty weight can be expressed as Equation (65).

$$W_{E_{New}} = W_E \left(1 + U_{EW}\right) \tag{65}$$

The maximum takeoff weight is fixed based in the initial sizing analysis which precedes and is therefore outside the scope of this study. Thus, MTOW it is expected to remain constant as the empty weight changes. As referenced in Appendix A.2 payload and crew weights are expected to be fixed requirements. Since all other parameters are fixed, an increase in empty weight via U_{EW} is expected to directly correspond to a decrease of fuel available for the same mission, as show in Equations (66) and (67).

$$W_0 = W_E \left(1 + U_{EW}\right) + W_{PL} + W_{Crew} + W_{Fuel} \left(1 - U_{EW}\right) \tag{66}$$

$$\Delta W_0 = W_E * \Delta U_{EW} - W_{Fuel} * \Delta U_{EW} \tag{67}$$

A.2.2 Treatment of fuel consumption changes

Based on force balance logic, fuel consumption for a given segment is expected to scale with the weight of the vehicle in that segment. Thus, in order to determine what the appropriate scaling factor for the fuel should be, the relationship between the individual segment weight and the more macro weights like MTOW and empty weight must be determined.

At the start of taxi out, the aircraft will be at its takeoff gross weight. For this mission, that will be the maximum takeoff gross weight of the vehicle. Thus, taxi out fuel will be scaled with any changes in W_0.

Since the fuel consumed for most segments will be proportional to the weight of the vehicle at that point, it would be convenient to have a functional form which scaled the fuel based off weight. Weight fractions, used in conceptual sizing analyses, are already conveniently in this format. Thus, it will be assumed for most segments that the weight fraction will remain constant. By knowing either the initial or final weight for the segment and the weight fraction from the deterministic sizing, the fuel consumed during that segment can be calculated.

Taxi out, takeoff and climb fuel consumption will scale with changes in maximum takeoff weight

$$W_{Fuel_{BeforeCruise}} \propto W_0 \tag{68}$$

Cruise fuel consumption will be calculated as a fallout of the initial and final cruise weights.

$$W_{Fuel_{Cruise}} = W_{initial} - W_{landing} \tag{69}$$

Descent fuel consumption will scale with the vehicle landing weight.

$$W_{Fuel_{Descent}} \propto W_{ZFW} + W_{AfterDescent} \propto W_{ZFW} \tag{70}$$

Landing fuel consumption will scale with vehicle landing weight. For the purpose of this analysis, landing weight will be assumed to be equal to the empty weight plus payload, reserves, taxi fuel, landing fuel, and crew.

$$W_{landing} = W_{EW} + W_{PL} + W_C + W_{Fuel_{AfterDescent}} \tag{71}$$

The zero-fuel weight ((W_{ZFW})) of the aircraft is the weight of the vehicle without any remaining fuel. This is equivalent to the weight of the empty vehicle plus crew and payload weights, as shown in Equation (72).

$$W_{ZFW} = W_E + W_{PL} + W_{Crew} \tag{72}$$

Substituting into Equation (72) into Equation (71) yields Equation (73).

$$W_{landing} = W_{ZFW} + W_{Fuel_{AfterDescent}} \tag{73}$$

Thus, it can be assumed that landing weight is proportional to the aircraft zero-fuel weight.

$$W_{landing} \propto W_{ZFW} \tag{74}$$

Reserve fuel is potentially complicated. Reserve fuel consumption will scale with total fuel weight. For rudimentary sizing analyses, reserve fuel is often calculated as a percent of the total fuel consumed during a sizing mission. In more detailed analyses, a reserve mission profile is specified and the fuel required is assessed to meet that mission. This mission is expected to be flown after descent, thus the fuel required for this mission will scale with the landing weight of the vehicle. The change of reserve fuel during the more detailed analyses conveniently mimics the behavior of other post-cruise fuel consumption. Thus, reserve fuel will be scaled with the empty weight of the vehicle.

Trapped fuel is often combined into the reserve fuel calculation. Unlike all the other post-cruise mission segments which change proportional to the empty weight of the vehicle, trapped fuel scales with the size of the fuel tanks. However, since this fuel will be on the order of 1% of the total fuel, this difference will be ignored for traceability, and trapped fuel will instead be incorporated into and treated as part of the reserve fuel calculation.

Later segments are proportional to zero fuel weight.

$$W_{Fuel_{taxiin}} \propto W_{ZFW} \tag{75}$$

$$W_{Fuel_{landing}} \propto (W_{ZFW} + W_{Fuel_{taxiin}}) \propto W_{ZFW} \tag{76}$$

$$W_{Fuel_{reserves}} \propto \left(W_{ZFW} + W_{Fuel_{taxiin}} + W_{Fuel_{landing}}\right) \propto W_{ZFW} \tag{77}$$

It has been assumed that non-cruise segments will not be significantly affected by changes in the drag prediction. This assumption was made to maintain some simplicity in the mathematics for the sake of the analysis; however, it is not strictly true. Drag may affect amount of fuel used during climb, descent, and the reserve mission. These effects will be much smaller than the effect on cruise and ignoring their impact should not impact the trends of this analysis.

Since U_{EW} will not affect MTOW, it will is expected to have no direct impact on fuel consumption during taxi out, takeoff, or climb mission segments.

Generally, all fuel consumed in pre-cruise segments will scale with maximum take-off weight (W_0) while fuel consumption in post-cruise mission will scale with landing weight.

A.2.2.1 Fuel MA

Increasing the post-sizing fuel of the aircraft is expected to incur a penalty on the empty weight of the vehicle. The amount of penalty per percent increase in fuel will be aircraft specific; however, the general form of the impact is easy to address. Once the penalty function has been determined and converted to a percent empty weight, the additional weight will affect the vehicle's empty weight in exactly the same manner as the empty weight uncertainty parameter (U_{EW}). However, the penalty will not degrade the aircraft's fuel as U_{EW} did, instead increasing the takeoff weight of the vehicle. Thus, the direct impact of ... Both the additional fuel and the weight penalty for that fuel will increase the MTOW of the aircraft.

$$W_0 = W_E\left(1 + penalty\left(M_{fuel}\right)\right) + W_{PL} + W_{Crew} + W_{Fuel}\left(1 + M_{fuel}\right) \tag{78}$$

$$\Delta W_0 = W_E\left(penalty\left(M_{fuel}\right)\right) + W_{Fuel}\left(M_{fuel}\right) \tag{79}$$

A.2.3 All impacts

A.2.3.1 Initial cruise weight

$$W_{initial} = W_0 - W_{Fuel_{taxiout}} - W_{Fuel_{takeoff}} - W_{Fuel_{climb}} \tag{80}$$

$$W_{initial} \propto W_0 \tag{81}$$

Since W_0 is fixed with regards to changes in uncertain variables, the aircraft fuel weight must adjust when the empty weight factor is non-zero. Rearranging Equation (54), yields Equation (82).

$$W_{Fuel} = W_0 - W_E - W_{PL} - W_{Crew} \tag{82}$$

$$W_{Fuel}\left(U_{EW}\right) = W_0 - W_E * \left(1 + U_{EW}\right) - W_{PL} - W_{Crew} \tag{83}$$

$$\Delta W_{Fuel}\left(U_{EW}\right) = -W_E * U_{EW} \tag{84}$$

The initial cruise weight is assumed to be proportional to MTOW. Thus, the new initial cruise weight will change proportionally to the change in MTOW, as shown in

Equation (85)

$$W_{initial_{New}} = W_{inital} * \frac{W_{0_{New}}}{W_0} \tag{85}$$

It was assumed that the maximum takeoff weight is not dependent on the uncertain variables. Thus, Equation (79) shows MTOW to be only dependent on the impact of the fuel addition mitigation action.

Substituting Equation (79) into Equation (85), yields Equation (86) where the initial cruise weight is dependent only upon the fuel addition mitigation action.

$$\begin{aligned} W_{initial_{New}} =& W_{inital} \\ & * \frac{W_E\left(1 + penalty\left(M_{fuel}\right)\right) + W_{PL} + W_{Crew} + W_{Fuel}\left(1 + M_{fuel}\right)}{W_0} \end{aligned} \tag{86}$$

new initial cruise weight, simplified

$$W_{initial_{New}} = W_{inital} * \left(1 + \frac{W_E}{W_0} * penalty\left(M_{fuel}\right) + \frac{W_{Fuel}}{W_0} * M_{fuel}\right) \tag{87}$$

percent change in intitial cruise weight

$$\%W_{initial} = \frac{W_E}{W_0} * penalty\left(M_{fuel}\right) + \frac{W_{Fuel}}{W_0} * M_{fuel} \tag{88}$$

Landing weight

$$W_{landing} \propto W_{ZFW} \tag{89}$$

Zero-fuel weight

$$W_{ZFW} = W_E\left(1 + U_{EW}\right)\left(1 + penalty\left(M_{fuel}\right)\right) + W_{PL} + W_{Crew} \tag{90}$$

Percent change in zero-fuel weight

$$\Delta W_{ZFW} = W_E * U_{EW} + W_E * penalty\left(M_{fuel}\right) + W_E * U_{EW} * penalty\left(M_{fuel}\right) \tag{91}$$

Since U_{EW} and the weight penalty from the fuel addition mitigation action are expected to be small, their product will be considered negligible.

$$U_{EW} * penalty\left(M_{fuel}\right) \approx 0 \tag{92}$$

Thus, the change in zero-fuel weight is

$$\Delta W_{ZFW} \approx W_E * U_{EW} + W_E * penalty\left(M_{fuel}\right) \tag{93}$$

and the percent change in zero-fuel weight is

$$\%W_{ZFW} \approx \frac{W_E * U_{EW} + W_E * penalty\left(M_{fuel}\right)}{W_E + W_{PL} + W_{Crew}} \tag{94}$$

Since landing weight it proportional to zero-fuel weight

$$W_{landing} \propto W_{ZFW} \tag{95}$$

$$\%W_{landing} = \%W_{ZFW} \approx \frac{W_E * U_{EW} + W_E * penalty\left(M_{fuel}\right)}{W_E + W_{PL} + W_{Crew}} \tag{96}$$

$$\%W_{landing} \approx \frac{W_E}{W_{ZFW}}\left(U_{EW} + penalty\left(M_{fuel}\right)\right) \tag{97}$$

A.2.3.2 Final Cruise Weight

The logic for the determining the new final cruise weight is identical to that of the landing weight. Thus, the details of the math will be omitted for brevity.

$$\%W_{final} = \%W_{ZFW} \approx \frac{W_E * U_{EW} + W_E * penalty\left(M_{fuel}\right)}{W_E + W_{PL} + W_{Crew}} \tag{98}$$

$$\%W_{final} \approx \frac{W_E * \left(U_{EW} + penalty\left(M_{fuel}\right)\right)}{W_{ZFW}} \tag{99}$$

$$\%W_{final} \approx \frac{W_E}{W_{ZFW}}\left(U_{EW} + penalty\left(M_{fuel}\right)\right) \tag{100}$$

A.2.4 Weight Impact on L/D

Solve for L/D cruise in terms of CL and CD cruise

$$\left(\frac{L}{D}\right)_{Cruise} = \left(\frac{C_L}{C_D}\right)_{Cruise} = \frac{C_{L_{Cruise}}}{C_{D_{Cruise}}} \tag{101}$$

$$C_{L_{Cruise}} = \frac{W_{Cruise}}{\frac{1}{2}\rho V^2 S} \tag{102}$$

$$W_{Cruise} = \frac{W_{inital} + W_{final}}{2} \tag{103}$$

Recall Equation (87)

$$W_{initial_{New}} = W_{inital} * \left(1 + \frac{W_E}{W_0} * penalty\left(M_{fuel}\right) + \frac{W_{Fuel}}{W_0} * M_{fuel}\right) \tag{87}$$

$$W_{final_{New}} = W_{final} \frac{W_E\left(1 + U_{EW}\right)\left(1 + penalty\left(M_{fuel}\right)\right) + W_{PL} + W_{Crew}}{W_{ZFW}} \tag{104}$$

Since both U_{EW} and $penalty\left(M_{fuel}\right)$ are small, it will be assume that their product is negligible. This yields Equation (105).

$$W_{final_{New}} \approx W_{final} * \left(1 + \frac{W_E}{W_{ZFW}} * \left(U_{EW} + penalty\left(M_{fuel}\right)\right)\right) \tag{105}$$

Substituting Equations (87) and (105) into Equation (103) yields the solution for the new cruise weight shown in Equation (106).

$$\begin{aligned} W_{Cruise_{New}} \approx & \frac{W_{inital}}{2} * \left(1 + \frac{W_E}{W_0} * penalty\left(M_{fuel}\right) + \frac{W_{Fuel}}{W_0} * M_{fuel}\right) \\ & + \frac{W_{final}}{2} * \left(1 + \frac{W_E}{W_{ZFW}} * \left(U_{EW} + penalty\left(M_{fuel}\right)\right)\right) \end{aligned} \tag{106}$$

Flight conditions and aircraft dimensions will remain constant for all analyses. Thus,

$$C_{L_{Cruise_{New}}} = \frac{W_{Cruise_{New}}}{\frac{1}{2}\rho V^2 S} = C_{L_{Cruise}} * \frac{W_{Cruise_{New}}}{W_{Cruise}} \tag{107}$$

$$C_{D_{Cruise}} = C_{D_{min}} + K(C_{L_{Cruise}} - C_{L_{min}})^2 \tag{108}$$

$$C_{D_{Cruise_{New}}} = C_{D_{min}} + K\left(C_{L_{Cruise}} * \frac{W_{Cruise_{New}}}{W_{Cruise}} - C_{L_{min}}\right)^2 \tag{109}$$

$$\left(\frac{L}{D}\right) = \frac{C_{L_{Cruise}}}{C_{D_{min}} + K(C_{L_{Cruise}} - C_{L_{min}})^2} \tag{110}$$

$$\left(\frac{L}{D}\right)_{wt_{New}} = \frac{C_{L_{Cruise}} * \frac{W_{Cruise_{New}}}{W_{Cruise}}}{C_{D_{min}} + K\left(C_{L_{Cruise}} * \frac{W_{Cruise_{New}}}{W_{Cruise}} - C_{L_{min}}\right)^2} \tag{111}$$

A.2.5 Drag

Also recall that the other uncertainty parameter, drag uncertainty, will have a large impact on cruise. The drag uncertainty will apply a direct scalar to the cruise drag. This will alter the lift-to-drag ratio from Equation (111) by adding the drag factor as in Equation (112).

$$\left(\frac{L}{D}\right)_{New} = \left(\frac{L}{D}\right)_{wt_{New}} * \frac{1}{(1+U_D)} \tag{112}$$

Equation (113) shows the new lift-to-drag ratio

$$\left(\frac{L}{D}\right)_{New} = \frac{C_{L_{Cruise}} * \frac{W_{Cruise_{New}}}{W_{Cruise}}}{(1+U_D) * \left(C_{D_{min}} + K\left(C_{L_{Cruise}} * \frac{W_{Cruise_{New}}}{W_{Cruise}} - C_{L_{min}}\right)^2\right)} \frac{1}{(1+U_D)} \tag{113}$$

A.3 Effect on Responses

A.3.1 Approach Speed

$$\frac{Vapp_{New}}{Vapp_{Det}} = \frac{1.3\sqrt{\frac{2}{\rho}\frac{W_{landing_{New}}}{S}\frac{1}{C_{L_{max}}}}}{1.3\sqrt{\frac{2}{\rho}\frac{W_{landing}}{S}\frac{1}{C_{L_{max}}}}} \tag{114}$$

Substituting

$$\frac{Vapp_{New}}{Vapp_{Det}} = \frac{1.3\sqrt{\frac{2}{\rho}\frac{W_{landing}(1+U_{EW})(1+penalty(M))}{S}\frac{1}{C_{L_{max}}}}}{1.3\sqrt{\frac{2}{\rho}\frac{W_{landing}}{S}\frac{1}{C_{L_{max}}}}} \tag{115}$$

Cancelling terms

$$\frac{Vapp_{New}}{Vapp_{Det}} = \sqrt{(1+U_{EW})\,(1+penalty\,(M))} \tag{116}$$

A.3.2 Range

$$\frac{R_{New}}{R_{Det}} = \frac{\frac{V}{C_t}\left(\frac{L}{D}\right)\ln\left(\frac{W_{inital_{New}}}{W_{final_{New}}}\right)}{\frac{V}{C_t}\left(\frac{L}{D}\right)\ln\left(\frac{W_{inital}}{W_{final}}\right)} \tag{117}$$

Substituting variables

$$\frac{R_{New}}{R_{Det}} = \frac{\frac{V}{C_t}\left(\frac{L}{D}\right)\frac{1}{(1+U_D)}\ln\left(\frac{W_{inital}(1+M)}{W_{final}*(1+U_{EW})(1+penalty(M))}\right)}{\frac{V}{C_t}\left(\frac{L}{D}\right)\ln\left(\frac{W_{inital}}{W_{final}}\right)} \tag{118}$$

Cancelling terms

$$\frac{R_{New}}{R_{Det}} = \frac{1}{(1+U_D)} \frac{\ln\left(\left(\frac{W_{inital}}{W_{final}}\right)\frac{(1+M)}{(1+U_{EW})(1+penalty(M))}\right)}{\ln\left(\frac{W_{inital}}{W_{final}}\right)} \tag{119}$$

$$\frac{R_{New}}{R_{Det}} = \frac{1}{(1+U_D)} \left(\frac{\ln\left(\frac{(1+M)}{(1+U_{EW})(1+penalty(M))}\right)}{\ln\left(\frac{W_{inital}}{W_{final}}\right)} + 1\right) \tag{120}$$

$$\Delta R = \frac{V}{C_t}\left(\frac{L}{D}\right)_{New} \ln\left(\frac{W_{inital_{New}}}{W_{final_{New}}}\right) - \frac{V}{C_t}\left(\frac{L}{D}\right)_{Det} \ln\left(\frac{W_{inital_{Det}}}{W_{final_{Det}}}\right) \tag{121}$$

$$\%R = \frac{\frac{V}{C_t}\left(\left(\frac{L}{D}\right)_{New}\ln\left(\frac{W_{inital_{New}}}{W_{final_{New}}}\right) - \left(\frac{L}{D}\right)_{Det}\ln\left(\frac{W_{inital_{Det}}}{W_{final_{Det}}}\right)\right)}{\frac{V}{C_t}\left(\frac{L}{D}\right)_{Det}\ln\left(\frac{W_{inital_{Det}}}{W_{final_{Det}}}\right)} \tag{122}$$

$$\%R = \frac{\left(\frac{L}{D}\right)_{New}}{\left(\frac{L}{D}\right)_{Det}} * \frac{\ln\left(\frac{W_{inital_{New}}}{W_{final_{New}}}\right)}{\ln\left(\frac{W_{inital_{Det}}}{W_{final_{Det}}}\right)} - 1 \tag{123}$$

A.4 Compliance

To test for compliance, M is set to 0 and the above equations are evaluated. This will reduce the complexity of previous equations.

Adding a constraint on the maximum approach speed of the vehicle to Equation (116) yields Equation (124).

$$\frac{Vapp_{req}}{Vapp_{Det}} \geq \sqrt{(1+U_{EW})\,(1+penalty\,(M))} \tag{124}$$

Removing the effects of mitigation actions on approach speed in Equation (124) yields Equation (125).

$$\frac{Vapp_{req}}{Vapp_{Det}} \geq \sqrt{(1+U_{EW})} \tag{125}$$

Modifying Equation (120) to include a minimum range yields Equation (126).

$$\frac{R_{req}}{R_{Det}} \leq \frac{1}{(1+U_D)} \left(\frac{\ln\left(\frac{(1+M)}{(1+U_{EW})(1+penalty(M))}\right)}{\ln\left(\frac{W_{inital}}{W_{final}}\right)} + 1\right) \tag{126}$$

When measuring compliance without mitigation actions, the range equation can be simplified to Equation (127)

$$\frac{R_{req}}{R_{Det}} \leq \frac{1}{(1+U_D)} \left(\frac{\ln\left(\frac{1}{1+U_{EW}}\right)}{\ln\left(\frac{W_{inital}}{W_{final}}\right)} + 1 \right) \tag{127}$$

The probability of compliance can be calculated by evaluating these equations over the uncertainty space and determining how frequently the performance meets the set constraints.

APPENDIX B

GOODNESS OF FIT METRICS FOR NEURAL NETWORKS

The following tables and figures contain the goodness of fit measures for the surrogate models used in Chapter 6. The data used to create these responses are based off the the model in Section 6.7. These surrogate models were generated using the Basic Regression Analysis for Integrated Neural Networks (BRAINN) [42], which also created these goodness of fit metrics.

Tables 15, 17, 19, 21, 23, 25, 27, 29, 31 and 33 show the basic statistical measures of accuracy of each response. Tables 16, 18, 20, 22, 24, 26, 28, 30, 32 and 34 contain detailed percentiles of the error distributions associated with each response model. Figures 80 to 89 show the distribution of the model fit error and model representation error as well as both actual and residual by predicted plots of the surrogate model.

Table 15: Block Fuel for Design Range ($BlockFuel_{DesRange}$) Statistics

Statistic	Value
Number of hidden nodes	5
R-square of Training Set	0.999375
R-square of Validation Set	0.999356
R-square of Test Set	N/A
Model Fit Error (μ)	0.000408014
Model Fit Error (σ)	0.346837
Model Representation Error (μ)	0.00449382
Model Representation Error (σ)	0.3517

Table 16: Block Fuel for Design Range ($BlockFuel_{DesRange}$) Quantiles

Quantile	Residual	Pct Error
0	-6785.63	-3.62717
0.5	-1692.04	-0.904458
2.5	-1234.67	-0.659978
10	-786.074	-0.420185
25	-420.539	-0.224794
50	-22.2281	-0.0118817
75	401.546	0.214641
90	829.226	0.443252
97.5	1352.72	0.72308
99.5	1838.38	0.982681
100	2949.24	1.57648

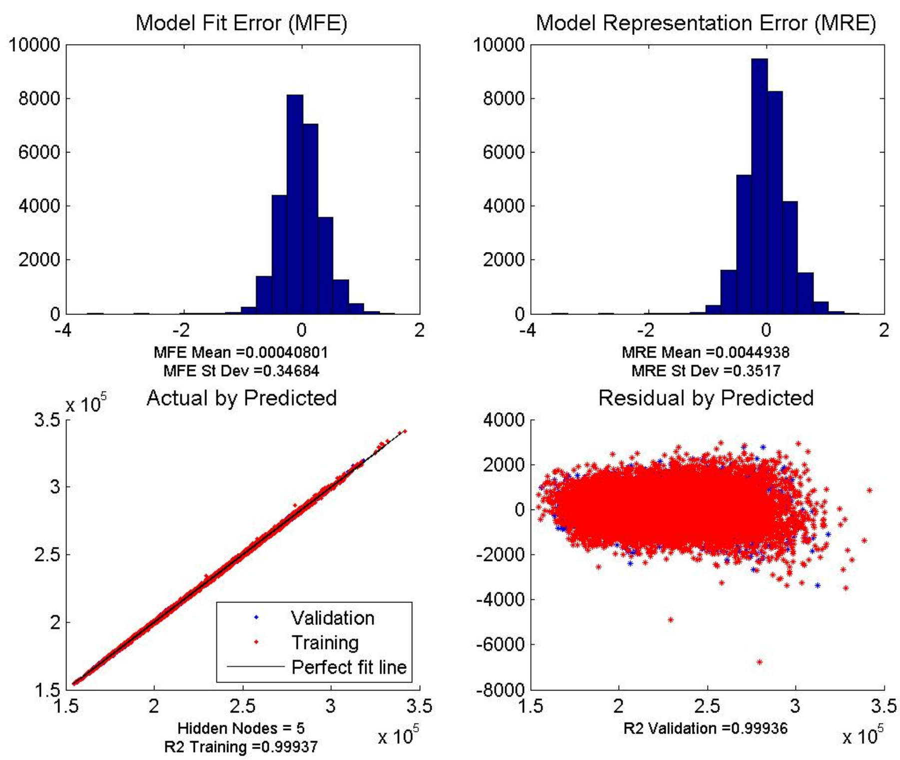


Figure 80: Goodness of Fit: Block Fuel for Design Range

Table 17: Block Fuel for Economic Mission Range ($BlockFuel_{Econ}$) Statistics

Statistic	Value
Number of hidden nodes	5
R-square of Training Set	0.995435
R-square of Validation Set	0.998758
R-square of Test Set	N/A
Model Fit Error (μ)	-0.000505608
Model Fit Error (σ)	0.548162
Model Representation Error (μ)	-0.00657038
Model Representation Error (σ)	0.28511

Table 18: Block Fuel for Economic Mission Range ($BlockFuel_{Econ}$) Quantiles

Quantile	Residual	Pct Error
0	-115970	-76.2956
0.5	-1332.57	-0.876685
2.5	-947.641	-0.623443
10	-559.914	-0.368362
25	-253.298	-0.166642
50	37.1544	0.0244435
75	287.399	0.189077
90	511.023	0.336197
97.5	796.773	0.524189
99.5	1045.65	0.687922
100	1863.01	1.22565

Table 19: Maximum Fuel Capacity (MaxFuel) Statistics

Statistic	Value
Number of hidden nodes	5
R-square of Training Set	0.999026
R-square of Validation Set	0.998991
R-square of Test Set	N/A
Model Fit Error (μ)	-0.000452863
Model Fit Error (σ)	0.403246
Model Representation Error (μ)	-0.000724607
Model Representation Error (σ)	0.409915

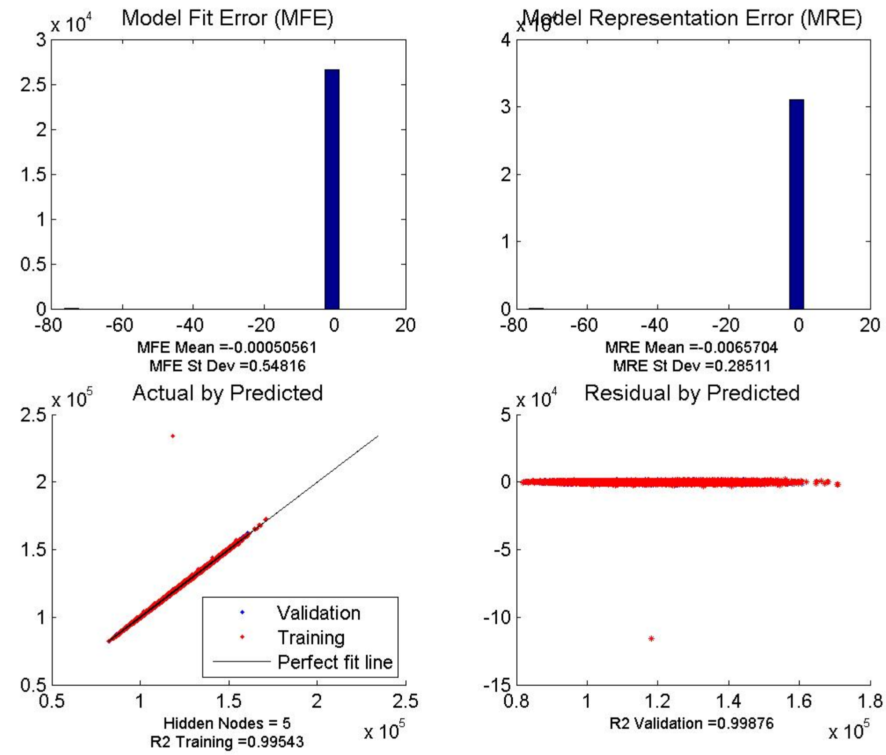


Figure 81: Goodness of Fit: Block Fuel for Economic Mission Range

Table 20: Maximum Fuel Capacity ($MaxFuel$) Quantiles

Quantile	Residual	Pct Error
0	-12536.9	-4.18029
0.5	-3273.64	-1.09156
2.5	-2234.1	-0.744935
10	-1432.21	-0.477555
25	-800.026	-0.26676
50	-55.7486	-0.0185888
75	754.33	0.251523
90	1546.61	0.515699
97.5	2501.73	0.834175
99.5	3474.37	1.15849
100	5927.41	1.97643

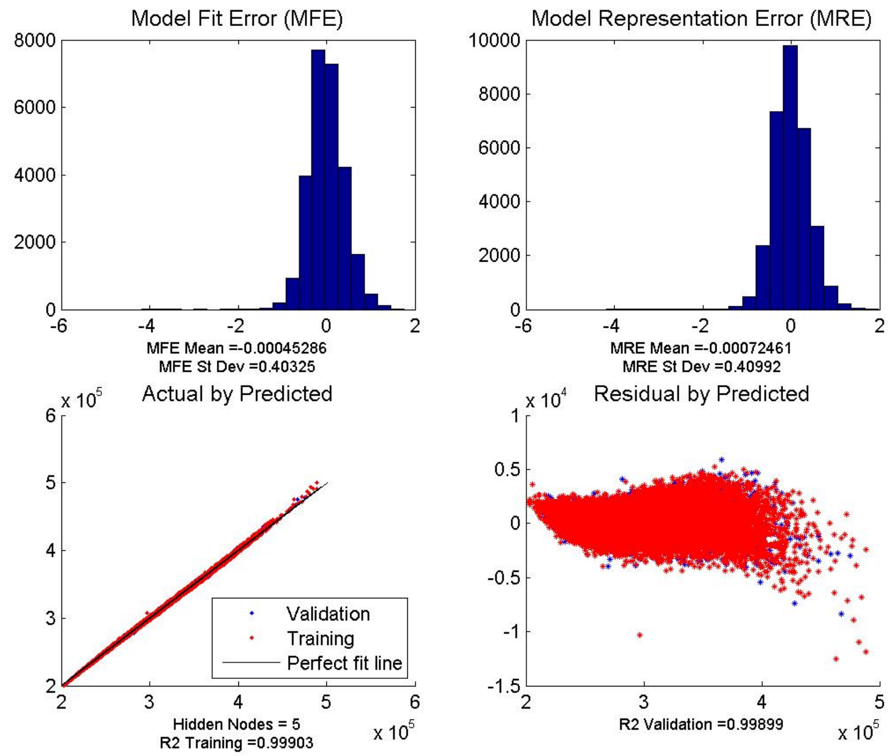


Figure 82: Goodness of Fit: Maximum Fuel Capacity

Table 21: Operating Empty Weight (OEW_u) Statistics

Statistic	Value
Number of hidden nodes	5
R-square of Training Set	0.998432
R-square of Validation Set	0.998325
R-square of Test Set	N/A
Model Fit Error (μ)	-0.00219417
Model Fit Error (σ)	0.530251
Model Representation Error (μ)	-0.0026155
Model Representation Error (σ)	0.536177

Table 22: Operating Empty Weight (OEW_u) Quantiles

Quantile	Residual	Pct Error
0	-9122.45	-4.38432
0.5	-3165.46	-1.52135
2.5	-2261.49	-1.08689
10	-1399.64	-0.672676
25	-726.581	-0.3492
50	14.5106	0.00697391
75	777.978	0.373902
90	1388.34	0.667246
97.5	2026.7	0.974047
99.5	2686.87	1.29133
100	4512.11	2.16855

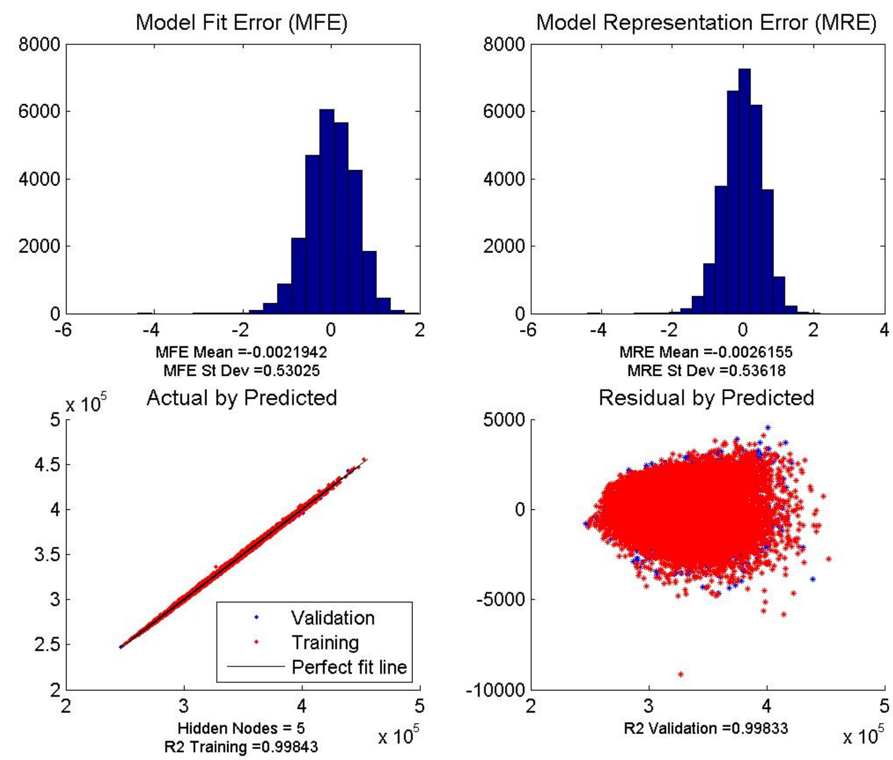


Figure 83: Goodness of Fit: Operating Empty Weight

Table 23: Ramp Weight ($RampWt$) Statistics

Statistic	Value
Number of hidden nodes	5
R-square of Training Set	0.998881
R-square of Validation Set	0.998797
R-square of Test Set	N/A
Model Fit Error (μ)	0.000956923
Model Fit Error (σ)	0.43256
Model Representation Error (μ)	-0.00435292
Model Representation Error (σ)	0.440997

Table 24: Ramp Weight ($RampWt$) Quantiles

Quantile	Residual	Pct Error
0	-14600.4	-3.67413
0.5	-4191.96	-1.05489
2.5	-3212.01	-0.80829
10	-2120.94	-0.533726
25	-1180.93	-0.297177
50	-68.3062	-0.017189
75	1118.54	0.281475
90	2252.72	0.566886
97.5	3521.51	0.886174
99.5	4786.37	1.20447
100	8350.02	2.10125

Table 25: Range at Design Payload ($Range$) Statistics

Statistic	Value
Number of hidden nodes	5
R-square of Training Set	0.999565
R-square of Validation Set	0.999549
R-square of Test Set	N/A
Model Fit Error (μ)	0.000235218
Model Fit Error (σ)	0.300397
Model Representation Error (μ)	-0.000797779
Model Representation Error (σ)	0.302675

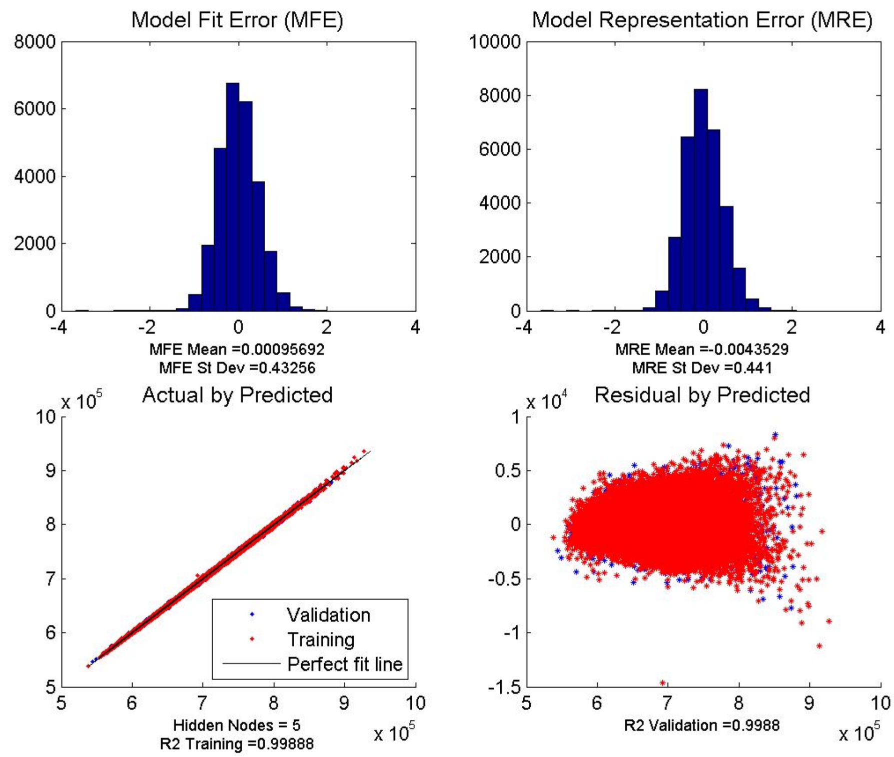


Figure 84: Goodness of Fit: Ramp Weight

Table 26: Range at Design Payload (*Range*) Quantiles

Quantile	Residual	Pct Error
0	-228.272	-2.75807
0.5	-83.5176	-1.00909
2.5	-56.5816	-0.683642
10	-32.4679	-0.39229
25	-13.6325	-0.164714
50	2.95887	0.0357502
75	16.5107	0.199489
90	27.5404	0.332754
97.5	41.9659	0.507049
99.5	60.6812	0.733175
100	154.964	1.87234

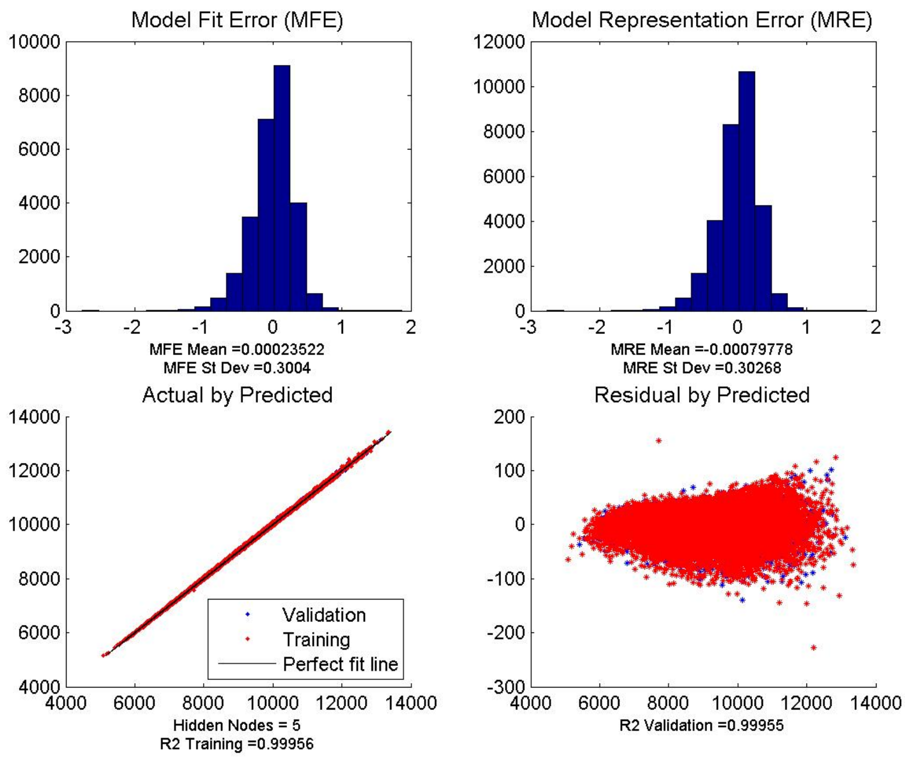


Figure 85: Goodness of Fit: Range at Design Payload

Table 27: Rate of Climb ($RateofClimb$) Statistics

Statistic	Value
Number of hidden nodes	5
R-square of Training Set	0.99964
R-square of Validation Set	0.999686
R-square of Test Set	N/A
Model Fit Error (μ)	-0.000143708
Model Fit Error (σ)	0.29485
Model Representation Error (μ)	-0.00334495
Model Representation Error (σ)	0.271658

Table 28: Rate of Climb ($RateofClimb$) Quantiles

Quantile	Residual	Pct Error
0	-292.365	-14.3583
0.5	-7.66057	-0.376219
2.5	-5.86593	-0.288082
10	-4.12499	-0.202583
25	-2.53001	-0.124251
50	-0.53937	-0.026489
75	1.84893	0.0908029
90	4.47373	0.21971
97.5	8.29931	0.407588
99.5	12.7002	0.623721
100	230.95	11.3422

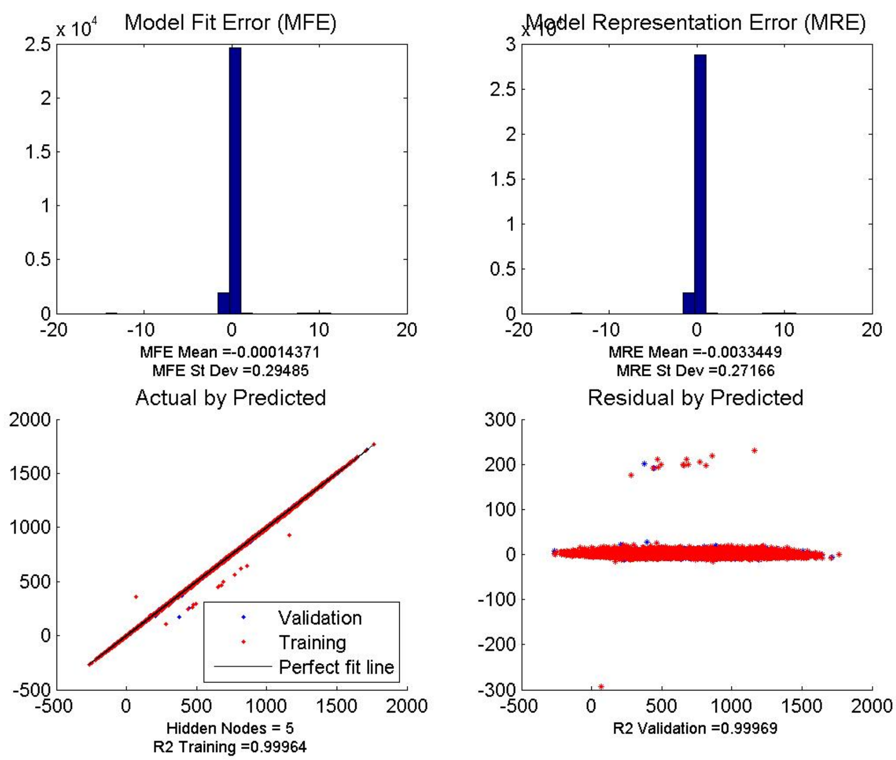


Figure 86: Goodness of Fit: Rate of Climb

Table 29: Span ($Span$) Statistics

Statistic	Value
Number of hidden nodes	5
R-square of Training Set	0.999328
R-square of Validation Set	0.999266
R-square of Test Set	N/A
Model Fit Error (μ)	-0.00182156
Model Fit Error (σ)	0.343902
Model Representation Error (μ)	-0.00303405
Model Representation Error (σ)	0.350375

Table 30: Span ($Span$) Quantiles

Quantile	Residual	Pct Error
0	-2.38835	-3.00165
0.5	-0.711365	-0.894034
2.5	-0.515933	-0.648418
10	-0.33591	-0.422167
25	-0.189394	-0.238028
50	-0.0128909	-0.0162012
75	0.179895	0.226089
90	0.35992	0.452343
97.5	0.554942	0.697443
99.5	0.739784	0.929751
100	1.16952	1.46984

Table 31: Takeoff Field Length ($TOFL$) Statistics

Statistic	Value
Number of hidden nodes	5
R-square of Training Set	0.95659
R-square of Validation Set	0.946081
R-square of Test Set	N/A
Model Fit Error (μ)	-1.31359E-05
Model Fit Error (σ)	1.96838
Model Representation Error (μ)	0.0215579
Model Representation Error (σ)	2.19783

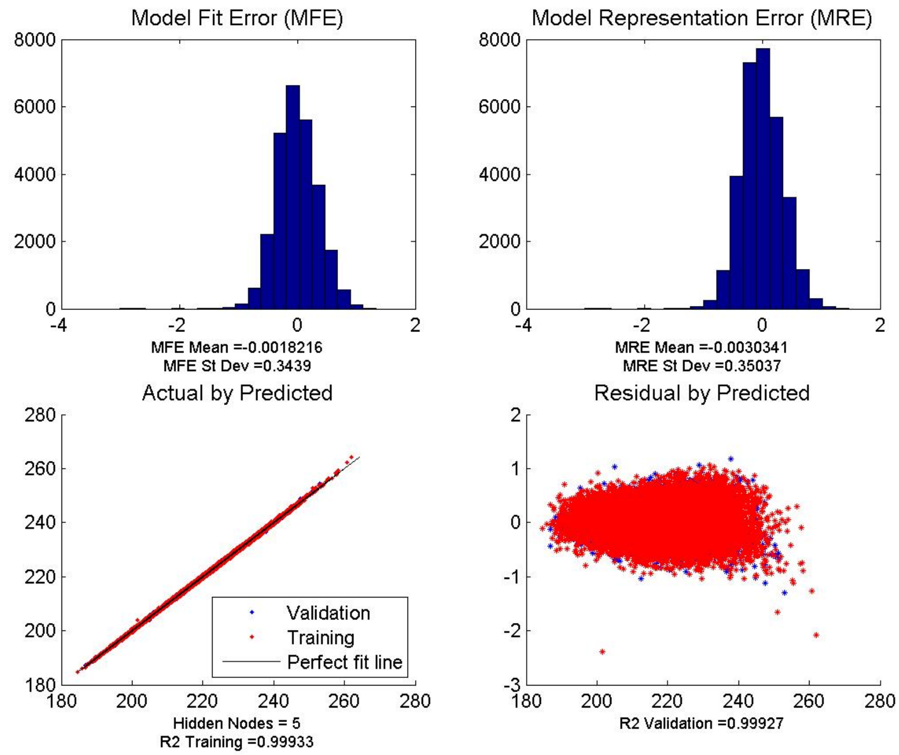


Figure 87: Goodness of Fit: Span

Table 32: Takeoff Field Length ($TOFL$) Quantiles

Quantile	Residual	Pct Error
0	-11076.6	-67.9461
0.5	-951.132	-5.83445
2.5	-725.456	-4.45011
10	-312.303	-1.91574
25	-116.813	-0.716557
50	10.6423	0.0652824
75	119.767	0.734679
90	307.108	1.88387
97.5	701.776	4.30485
99.5	1019.95	6.25659
100	2542.44	15.5959

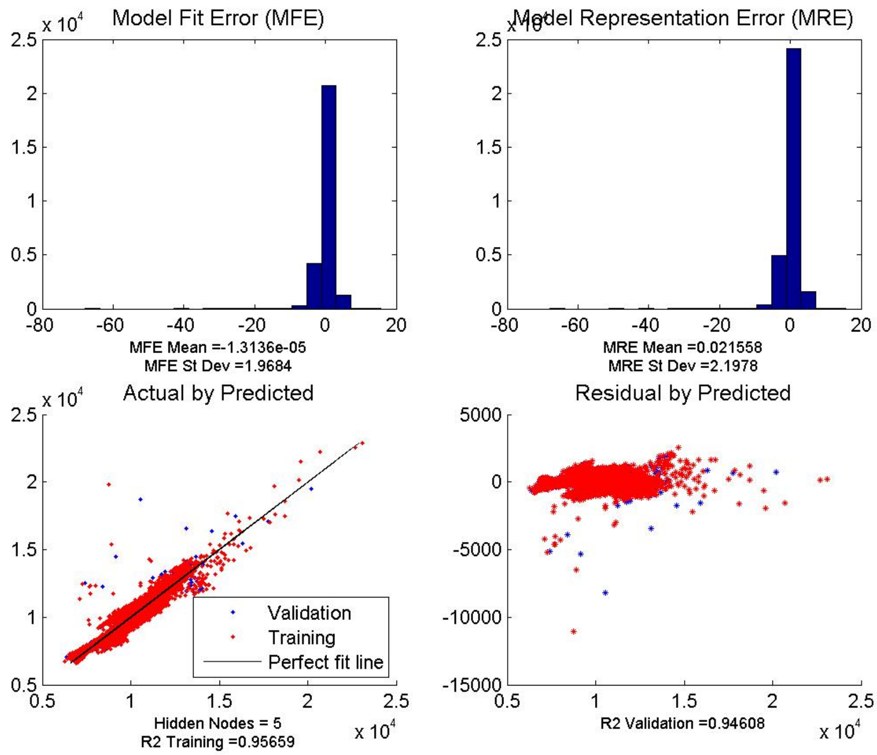


Figure 88: Goodness of Fit: Takeoff Field Length

Table 33: Approach Speed (V_{App}) Statistics

Statistic	Value
Number of hidden nodes	5
R-square of Training Set	0.99975
R-square of Validation Set	0.999755
R-square of Test Set	N/A
Model Fit Error (μ)	-0.000575597
Model Fit Error (σ)	0.224677
Model Representation Error (μ)	-0.00103335
Model Representation Error (σ)	0.223749

Table 34: Approach Speed (V_{App}) Quantiles

Quantile	Residual	Pct Error
0	-0.328544	-1.14078
0.5	-0.161621	-0.561183
2.5	-0.122281	-0.424587
10	-0.0800994	-0.278123
25	-0.0432939	-0.150326
50	-0.00247068	-0.00857874
75	0.0412205	0.143127
90	0.0823615	0.285977
97.5	0.133702	0.464244
99.5	0.184358	0.640133
100	0.368989	1.28121

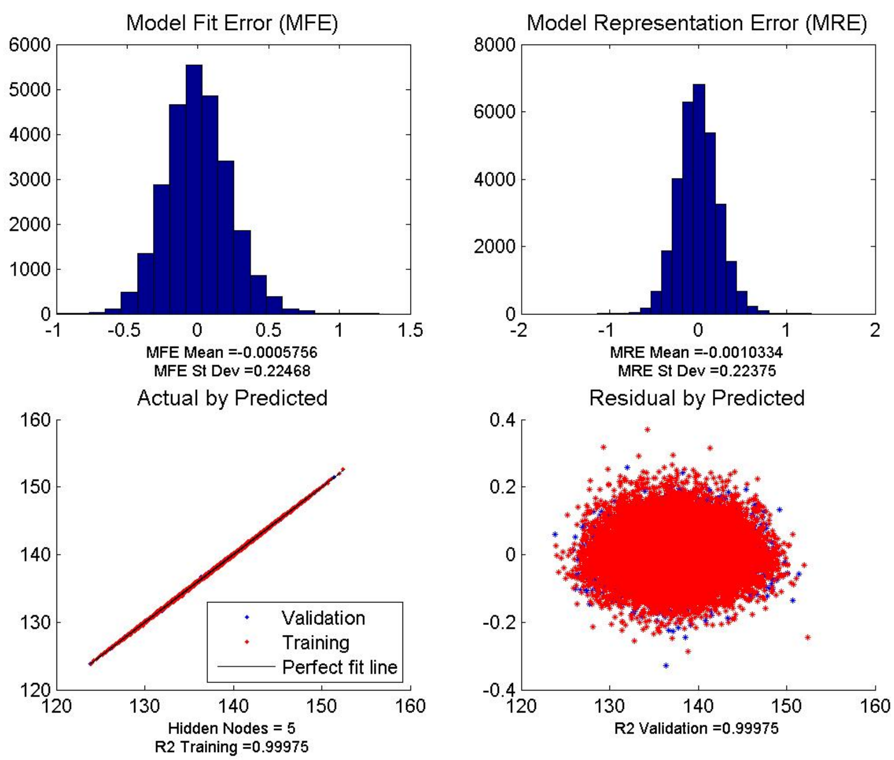


Figure 89: Goodness of Fit: Approach Speed

APPENDIX C

CODE FOR ARMOUR UNCERTAINTY QUANTIFICATION & MANAGEMENT ENVIRONMENT

This appendix contains the logic behind the actual code implemented for the uncertainty quantification and management environment behind the ARMOUR methodology. The method used to construct this environment is described in Section 5.9, and the specific implementation used to generate results in Chapter 7 is described in Section 6.9. This uncertainty Quantification and management environment was constructed and tested in MATLAB 2014a.

```
% Run_Opt_UQM
clear
start_time = datestr(now);
fprintf(1,'Start Time is %s\n',start_time);

% Disable parallel pools
ps = parallel.Settings;
ps.Pool.AutoCreate = false;

% General Settings
optimization_method = 'GA';      % Options: GA, DIRect
distribution_type = 'uniform';      % Options:tri-centered, ...
    Rayleigh, normal, uniform, marg_set_distrib
Num_Reliability_Cases = 10000; % 5000;
Num_Constr = 6;
GA_PopulationSize = 100;
Mitigation_Method = 'none';      %'optimize'; 'none', ...
    'random_search', 'sweep', '1D', 'hybrid1D'
default_Mitigation_Method = 'hybrid1D';
num_MA_levels = 5;      % number of levels to divide a mitigation ...
    action into for sweep
random_search_cases = 10 * num_MA_levels;      % number of ...
    randomly selected mitigation cases to use for random_search

StartingPoint;

```

```
Reliability_Goal_Array = [0.75; 0.76; 0.77; 0.78; 0.79; 0.8; ...
    0.81; 0.82; 0.83; 0.84; 0.85; 0.86; 0.87; 0.88; 0.89; 0.9; ...
    0.91; 0.92; 0.93; 0.94; 0.95; 0.96; 0.97; 0.98; 0.99; 1];
%Reliability_Goal_Array = [0.80];

Mitigation_Wt_Array = [0.0];

%Success_Goal_Array = [0.84; 0.86; 0.88; 0.94; 0.96; 0.98];
SSuccess_Goal_Array = [0.0];
%Success_Goal_Array = [0.75; 0.76; 0.77; 0.78; 0.79; 0.8; 0.81; ...
    0.82; 0.83; 0.84; 0.85; 0.86; 0.87; 0.88; 0.89; 0.9; 0.91; ...
    0.92; 0.93; 0.94; 0.95; 0.96; 0.97; 0.98; 0.99; 1];

% ------  File Name  ------
file_iteration = 1750;
sub_type = [optimization_method, num2str(GA_PopulationSize)];
sub_type = [sub_type, '_C', num2str(Reliability_Goal_Array(1)), ...
    '-', ...
    num2str(Reliability_Goal_Array(size(Reliability_Goal_Array,1)))];
sub_type = [sub_type, '_S', num2str(Success_Goal_Array(1)), '-', ...
    num2str(Success_Goal_Array(size(Success_Goal_Array,1)))];
if strcmp(Mitigation_Method, 'none')
     mitigation_info = [sub_type, '_def'];
     useM = default_Mitigation_Method;
end
create_out_file;

%------------------------------
% GA Options
options_GA = gaoptimset('Display', 'iter');      % iter
options_GA = gaoptimset(options_GA, 'PopulationSize', ...
    GA_PopulationSize);
%------------------------------
% fmincon Options
options = optimset('UseParallel', 'never');
options = optimset(options, 'Display', 'off');
options = optimset(options, 'Algorithm', 'sqp');
options = optimset(options, 'TolCon', 1e-5, 'TolFun', 1e-5);

% =====================================
% Setup
tic
total_time = toc;

% bound for gradient-based optimization
Aeq = [];
beq = [];
A = [eye(size(X0,1));-1*eye(size(X0,1))];
b = [UB;LB];
bounds = [LB,UB];

% Declaring variables
Reliability_goal_sweep_count = size(Reliability_Goal_Array,1);
Mitigation_Sweep_count       = size(Mitigation_Wt_Array,1);
```

```
Success_goal_sweep_count      = size(Success_Goal_Array,1);
runs_to_perform = Reliability_goal_sweep_count * ...
    Mitigation_Sweep_count * Success_goal_sweep_count;
basic_variable_declaration;

fprintf(1, '\n\n');
fprintf(1, '================================================\n');
fprintf(1, '============ Running %i scenarios ============\n', ...
    runs_to_perform);
fprintf(1, 'Start Time is \t%s\n', datestr(now));
fprintf(1, 'Saving to file %s.\n', out_file);

% Establish uncertainty cases
S_Cases = Random_Cases_Setup(Num_Reliability_Cases, Num_S_Vars, ...
    distribution_type);
remember_Mitigation_Method = Mitigation_Method;
WriteOutput_setup;

% ======================================
% Main Loop
for j_main = 1:Reliability_goal_sweep_count
    Reliability_goal = Reliability_Goal_Array(j_main);

    for l_main = 1:Success_goal_sweep_count
        Success_goal = Success_Goal_Array(l_main);

        for k_main = 1:Mitigation_Sweep_count
            i_main = (j_main - 1)*Mitigation_Sweep_count * ...
                Success_goal_sweep_count + (l_main - ...
                1)*Mitigation_Sweep_count + k_main;
            if i_main > 1
                X0 = x;
            end

            Mitigation_Weighting = Mitigation_Wt_Array(k_main);

            Mitigation_Method = remember_Mitigation_Method;
            if ((Mitigation_Weighting > 0 || Success_goal > 0) && ...
                strcmp(Mitigation_Method, 'none'))
                Mitigation_Method = default_Mitigation_Method;
                fprintf(1,'Activating Mitigation_Method = %s\n', ...
                    Mitigation_Method);
            end
            temp_remember_Mitigation_Method = Mitigation_Method;

            fprintf(1, '======================================\n');
            fprintf(1, 'Run %i of %i\n', i_main, runs_to_perform);
            fprintf(1, 'Reliability goal  = %5.3f,  Success goal ...
                = %5.3f,  Mitigation Wt      = %5.3f\n', ...
                Reliability_goal, Success_goal, Mitigation_Weighting);
            fprintf(1, 'Num Reliab Cases  = %5i,  Constraints  = ...
                %5i,  Mitigation_Method = %s\n', ...
                Num_Reliability_Cases, Num_Constr, Mitigation_Method);
            if strcmp(optimization_method, 'DIRect')
```

```
            fprintf(1, 'DIRect evals      = %5i\n', ...
                DIRect_maxevals);
        elseif strcmp(optimization_method, 'GA')
            fprintf(1, 'GA_PopulationSize = %5i\n', ...
                GA_PopulationSize);
        end
        fprintf(1, '%s\tStart Loop\n',datestr(now));
        StartingPoint

        % Functions to optimize
        fitness_GA    = @(x) fmincon_interface_EXS(           ...
            x, S_Cases, Reliability_goal, Num_Constr, ...
            Mitigation_Weighting, Mitigation_Method, ...
            num_MA_levels, random_search_cases);
        constraint_GA = @(x) ...
            fmincon_interface_EXS_constraint(x, S_Cases, ...
            Reliability_goal, Success_goal, Num_Constr, ...
            Mitigation_Weighting, Mitigation_Method, ...
            num_MA_levels, random_search_cases);

        % Run global optimizer
        fprintf(1, '%s\tRunning %s\n',datestr(now), ...
            optimization_method);
        if strcmp(optimization_method, 'DIRect')
            [minval,xatmin] = ...
                Direct(to_optimize,bounds,DIRect_opts,S_Cases, ...
                Reliability_goal, Num_Constr, ...
                Mitigation_Weighting, Mitigation_Method);
        elseif strcmp(optimization_method, 'GA')
            [xatmin, minval,EXITFLAG,OUTPUT] = ga(fitness_GA, ...
                size(X0,1), [], [], [], [], LB, UB, ...
                constraint_GA, options_GA);
        elseif strcmp(optimization_method, 'skip')
            xatmin = X0;
        end
        fprintf(1, '%s\t%s Finished\n',datestr(now), ...
            optimization_method);

        % Calculate final values
        if strcmp(Mitigation_Method, 'none')
            Mitigation_Method = default_Mitigation_Method;
        end
        [AA_final_Block_Fuel, AA_final_Reliability, ...
            AA_final_Reliability_MA, S_fail, S_fail_MA, ...
            Reliability_Individual, Expected_BlockFuel_Des, ...
            Expected_BlockFuel_DOC] = Evaluate_X_S(xatmin, ...
            S_Cases, Reliability_goal, Mitigation_Method, ...
            Num_Constr, num_MA_levels, random_search_cases,1);
        Mitigation_Method = temp_remember_Mitigation_Method;

        % Save global results
        xatmin_array(:,i_main) = xatmin;
        x = xatmin;
        AA_Global_Reliability(i_main)     = AA_final_Reliability;
```

```
            AA_Global_Reliability_MA(i_main)  = ...
               AA_final_Reliability_MA;
            AA_Global_Reliability_Individual(:,i_main)      = ...
               Reliability_Individual;
            AA_Global_Block_Fuel(i_main)       = AA_final_Block_Fuel;
            AA_Global_Expected_BF_Des(i_main) = ...
               Expected_BlockFuel_Des;
            AA_Global_Expected_BF_DOC(i_main) = ...
               Expected_BlockFuel_DOC;
            if strcmp(optimization_method, 'GA')
                function_calls_global_array(i_main)     = ...
                    OUTPUT.funccount;
            end

            % Run gradient-based optimizer
            fprintf(1, '%s\tRunning fmincon\n',datestr(now));
            [x,FVAL,EXITFLAG,OUTPUT] = fmincon(fitness_GA, x, A, ...
               b, Aeq, beq, LB, UB, constraint_GA, options);
            fprintf(1, '%s\tFinished fmincon\n',datestr(now));

            % Final Values
            fprintf(1, '%s\tCalculating Final ...
               Values\n',datestr(now));
            if strcmp(Mitigation_Method, 'none')
                Mitigation_Method = default_Mitigation_Method;
            end
            [AA_final_Block_Fuel, AA_final_Reliability, ...
               AA_final_Reliability_MA, ~, ~, ...
               Reliability_Individual, Expected_BlockFuel_Des, ...
               Expected_BlockFuel_DOC, Expected_Range, ...
               Expected_TOFL, Expected_VAPP, ...
               Expected_RateofClimb, Expected_Span, ...
               Expected_RampWt, Expected_MaxFuel, Expected_OEW, ...
               Expected_Fuel_Used] = Evaluate_X_S(x, S_Cases, ...
               Reliability_goal, Mitigation_Method, Num_Constr, ...
               num_MA_levels, random_search_cases,1);
            Mitigation_Method = temp_remember_Mitigation_Method;

            % Save results
            fprintf(1, '%s\tSaving Results\n',datestr(now));
            x_array(:,i_main) = x;
            AA_array_Reliability(i_main) = AA_final_Reliability;
            AA_array_Reliability_MA(i_main) = ...
               AA_final_Reliability_MA;
            AA_array_Reliability_Individual(:,i_main) = ...
               Reliability_Individual;
            AA_array_Block_Fuel(i_main) = AA_final_Block_Fuel;
            AA_array_Expected_BF_Des(i_main) = ...
               Expected_BlockFuel_Des;
            AA_array_Expected_BF_DOC(i_main) = ...
               Expected_BlockFuel_DOC;
            EXITFLAG_array(i_main) = EXITFLAG;
            iterations_array(i_main) = OUTPUT.iterations;
            function_calls_array(i_main) = OUTPUT.funcCount;
```

```
            Reliability_goal_out_array(i_main) = Reliability_goal;
            Success_goal_out_array(i_main) = Success_goal;
            Mitigation_Weight_out_array(i_main) = ...
               Mitigation_Weighting;

            % Time per call
            total_time = total_time + toc;
            time_per_function_call = toc / OUTPUT.funcCount;
            tic;

            %PrintResults
            PrintResults;
            fprintf(1, '%s\tWrite Outputs to file: %s\n', ...
               datestr(now), out_file);
            WriteOutput_case;
        end
    end
end

fprintf(1, '====================================\n');
PrintOverall;
fprintf(1,'Start Time was %s\n',start_time);
fprintf(1,'End Time is    %s\n',datestr(now));

fid = fopen(out_file,'a+');
fprintf(fid,'\nStart Time,%s\nEnd Time,%s\n', start_time, ...
   datestr(now));
fclose(fid);
fprintf(1, '==============================================\n');
```

```
% Run_MCS
clear
start_time = datestr(now);
fprintf(1,'Start Time is %s\n',start_time);

% General Settings
x_cases = 5000;
optimization_method = 'MCS';        % Options: GA, DIRect
distribution_type = 'uniform';      % Options:tri-centered, ...
   Rayleigh, normal, uniform, marg_set_distrib
Num_Reliability_Cases = 5000; % 50000;
Num_Constr = 6;
Mitigation_Method = '1D';        %'optimize'; 'none', ...
   'random_search', 'sweep', '1D', 'hybrid1D'
num_MA_levels = 5;       % number of levels to divide a mitigation ...
   action into for sweep
random_search_cases = 10 * num_MA_levels;       % number of ...
   randomly selected mitigation cases to use for random_search

StartingPoint;

% ------  File Name  ------
```

```
file_iteration = 2011;
sub_type = ['x', int2str(x_cases)];
create_out_file;

% =====================================
% Setup
tic
total_time = toc;

% Declaring variables
runs_to_perform = x_cases;
basic_variable_declaration;
x_array = rand(size(X0,1), runs_to_perform);
Reliability_goal = 0;
Mitigation_Weighting = 0;

fprintf(1, '\n\n');
fprintf(1, '==============================================\n');
fprintf(1, '============ Running %i scenarios ============\n', ...
    runs_to_perform);
fprintf(1,'Start Time is \t%s\n',datestr(now));
fprintf(1,'Saving to file %s.\n',out_file);

% Establish uncertainty cases
rng(0);
S_Cases = Random_Cases_Setup(Num_Reliability_Cases, Num_S_Vars, ...
    distribution_type);
Set_S_matrix;
rng('shuffle');

check_after = ceil(x_cases/100);
WriteOutput_setup;

% =====================================
% Main Loop
for i_main = 1:x_cases
    x = x_array(:, i_main);
    if (rem(i_main,check_after) == 0)
        fprintf(1,'%s \tRunning Case %i of %i\n', datestr(now), ...
            i_main, x_cases);
    end

    % Calculate performance
    [AA_final_Block_Fuel, AA_final_Reliability, ...
        AA_final_Reliability_MA, ~, ~, Reliability_Individual, ...
        Expected_BlockFuel_Des, Expected_BlockFuel_DOC, ...
        Expected_Range, Expected_TOFL, Expected_VAPP, ...
        Expected_RateofClimb, Expected_Span, Expected_RampWt, ...
        Expected_MaxFuel, Expected_OEW, Expected_Fuel_Used] = ...
        Evaluate_X_S(x, S_Cases, 0.8, Mitigation_Method, ...
            Num_Constr, num_MA_levels, random_search_cases);

    % Save results
    %x_array(:,i_main) = x;
```

```
    AA_array_Reliability(i_main)      = AA_final_Reliability;
    AA_array_Reliability_MA(i_main)   = AA_final_Reliability_MA;
    AA_array_Reliability_Individual(:,i_main)      = ...
        Reliability_Individual;
    AA_array_Block_Fuel(i_main)       = AA_final_Block_Fuel;
    AA_array_Expected_BF_Des(i_main)  = Expected_BlockFuel_Des;
    AA_array_Expected_BF_DOC(i_main)  = Expected_BlockFuel_DOC;

    %PrintResults
    WriteOutput_case;
end

PrintOverall;
fprintf(1, 'Start Time was %s\n', start_time);
fprintf(1, 'End Time is    %s\n', datestr(now));

fid = fopen(out_file,'a+');
fprintf(fid, '\nStart Time,%s\nEnd Time,%s\n', start_time, ...
    datestr(now));
fclose(fid);
fprintf(1, '===============================================\n');
```

```
% Run_DOE.m
clear
start_time = datestr(now);
fprintf(1,'Start Time is %s\n',start_time);

% Read in design of experiments
doe_filename = 'doe_baseline_opt.txt';
delimiterIn = '\t';
headerlinesIn = 1;
DoeTable = importdata(doe_filename,delimiterIn,headerlinesIn);
x_cases = size(DoeTable.data,1);

first_col = DoeTable.colheaders(1);
if strcmp (first_col, 'Case') || strcmp (first_col, 'CASE')
    has_case_col = 1;
elseif strcmp (first_col, 'TWR')
    has_case_col = 0;
else
    fprintf(1,'First column name of %s is not ...
        recognized\n',first_col);
    break
end

% General Settings
optimization_method = 'DoE';         % Options: GA, DIRect
distribution_type = 'uniform';
Num_Reliability_Cases = 10000; %5000;
Num_Constr = 6;
Mitigation_Method = 'hybrid1D'; %random_search, 'optimize', ...
    'sweep', '1D', 'hybrid1D'
```

```
num_MA_levels = 20;      % number of levels to divide a mitigation ...
    action into for sweep
random_search_cases = 40 * num_MA_levels;       % number of ...
    randomly selected mitigation cases to use for random_search

StartingPoint;

% ------  File Name  ------
file_iteration = 3000;
sub_type = doe_filename;
create_out_file;

% =====================================
% Setup
tic
total_time = toc;

% Declaring variables
runs_to_perform = x_cases;
basic_variable_declaration;
Reliability_goal = 0.0;
Mitigation_Weighting = 0;

% Establish uncertainty cases
rng(0);
S_Cases = Random_Cases_Setup(Num_Reliability_Cases, Num_S_Vars, ...
    distribution_type);
Set_S_matrix;
rng('shuffle');

check_after = ceil(x_cases/100);
WriteOutput_setup;

% =====================================
% Main Loop
for i_main = 1:x_cases
    if (rem(i_main,check_after) == 0)
        fprintf(1,'%s \tRunning Case %i of %i\n', datestr(now), ...
            i_main, x_cases);
    end

    % Read design
    i_var = 0 + has_case_col;
    TWR = DoeTable.data(i_main,i_var+1);    i_var=i_var+1;
    WSR = DoeTable.data(i_main,i_var+1);    i_var=i_var+1;
    AR  = DoeTable.data(i_main,i_var+1);    i_var=i_var+1;
    EWMARG  = DoeTable.data(i_main,i_var+1);    i_var=i_var+1;
    FCDSUB  = DoeTable.data(i_main,i_var+1);    i_var=i_var+1;
    FACT    = DoeTable.data(i_main,i_var+1);    i_var=i_var+1;
    LapseRate   = DoeTable.data(i_main,i_var+1);    i_var=i_var+1;
    FPR = DoeTable.data(i_main,i_var+1);    i_var=i_var+1;
    OPR = DoeTable.data(i_main,i_var+1);    i_var=i_var+1;

    % Convert design to values from 0 to 1
```

```
    unscale_x;

    % Set values for records
    for i_var = 1:9
        x_scaled(i_main,i_var) = DoeTable.data(i_main, i_var + ...
            has_case_col);
    end

    % Calculate performance
    [AA_final_Block_Fuel, AA_final_Reliability, ...
       AA_final_Reliability_MA, ~, ~, Reliability_Individual, ...
       Expected_BlockFuel_Des, Expected_BlockFuel_DOC, ...
       Expected_Range, Expected_TOFL, Expected_VAPP, ...
       Expected_RateofClimb, Expected_Span, Expected_RampWt, ...
       Expected_MaxFuel, Expected_OEW, Expected_Fuel_Used] = ...
        Evaluate_X_S(x, S_Cases, Reliability_goal, ...
            Mitigation_Method, Num_Constr, num_MA_levels, ...
            random_search_cases);

    % Save results
    x_array(:,i_main) = x;
    AA_array_Reliability(i_main)     = AA_final_Reliability;
    AA_array_Reliability_MA(i_main)  = AA_final_Reliability_MA;
    AA_array_Reliability_Individual(:,i_main) = ...
       Reliability_Individual;
    AA_array_Block_Fuel(i_main)      = AA_final_Block_Fuel;
    AA_array_Expected_BF_Des(i_main) = Expected_BlockFuel_Des;
    AA_array_Expected_BF_DOC(i_main) = Expected_BlockFuel_DOC;

    %PrintResults
    WriteOutput_case;
end

PrintOverall;
fprintf(1,'Start Time was %s\n',start_time);
fprintf(1,'End Time is    %s\n',datestr(now));

fid = fopen(out_file,'a+');
fprintf(fid,'\nStart Time,%s\nEnd ...
    Time,%s\n',start_time,datestr(now));
fclose(fid);
fprintf(1, '==============================================\n');
```

```
%basic_variable_declaration.m

xatmin_array                     = zeros(size(X0, 1), ...
    runs_to_perform);
AA_Global_Reliability            = zeros(1, runs_to_perform);
AA_Global_Reliability_MA         = zeros(1, runs_to_perform);
AA_Global_Block_Fuel             = zeros(1, runs_to_perform);
AA_Global_Expected_BF_Des        = zeros(1, runs_to_perform);
AA_Global_Expected_BF_DOC        = zeros(1, runs_to_perform);
```

```
AA_Global_Reliability_Individual = zeros(Num_Constr, ...
    runs_to_perform);
Reliability_goal_out_array       = zeros(runs_to_perform, 1);
Mitigation_Weight_out_array      = zeros(runs_to_perform, 1);
Success_goal_out_array           = zeros(runs_to_perform, 1);

x_array                          = zeros(size(X0,1), ...
    runs_to_perform);
x_scaled                         = zeros(size(X0,1), ...
    runs_to_perform);
AA_array_Reliability             = zeros(1, runs_to_perform);
AA_array_Reliability_MA          = zeros(1, runs_to_perform);
AA_array_Block_Fuel              = zeros(1, runs_to_perform);
AA_array_Expected_BF_Des         = zeros(1, runs_to_perform);
AA_array_Expected_BF_DOC         = zeros(1, runs_to_perform);
AA_array_Reliability_Individual  = zeros(Num_Constr, ...
    runs_to_perform);
EXITFLAG_array                   = zeros(1, runs_to_perform);
iterations_array                 = zeros(1, runs_to_perform);
function_calls_array             = zeros(1, runs_to_perform);
function_calls_global_array      = zeros(1, runs_to_perform);
```

```
% calculate_MA_penalty.m

if MTOW_MA > 0
    OWE_delta = 0.0002608 .* MTOW_MA .^2  - 0.0063233 .* MTOW_MA;
else
    OWE_delta = 0;
end

OWE_delta = OWE_delta + 85000/7 .* (u_CLLDM - u_CLLDM_base) .^2 + ...
    3250/7 .* (u_CLLDM - u_CLLDM_base) - 3/7;

if OWE_delta < 0
    OWE_delta = 0;
end
```

```
% constraints.m

constr_Range = 7530;         %6800;  % 7530;
constr_TOFL  = 11000;        %12000; % 11000;
constr_VAPP  = 145;          % 140;
constr_RateofClimb  = 300;   % 300;
constr_Span = 215;           %230;   % 215;
constr_Fuel = 0;

case_constraints = zeros(Num_Reliability_Cases, Num_Constr);

% Evaluate surrogates
surrogates_777_X_2014_08_19

```

```
case_constraints(:, 1) = (Range(:) - constr_Range)    /constr_Range;
case_constraints(:, 2) = (constr_TOFL  - TOFL(:))     /constr_TOFL;
case_constraints(:, 3) = (constr_VAPP  - VAPP(:))     /constr_VAPP;
case_constraints(:, 4) = (RateofClimb(:) - constr_RateofClimb)    ...
    /constr_RateofClimb;
case_constraints(:, 5) = (constr_Span  - Span(:))     /constr_Span;
case_constraints(:, 6) = ((MaxFuel(:) - Fuel_Used(:)) - ...
    constr_Fuel) ./ MaxFuel(:);
if MTOW_MA <= 0
    case_constraints(:, 6) = 1;
end
```

```
%create_out_file.m
out_folder = 'results\';
run_type = [optimization_method, '_'];
rel_info = ['R', int2str(Num_Reliability_Cases), '_'];

if ~exist('useM','var')
    useM = Mitigation_Method;
end
mitigation_info = ['_', useM];

if (strcmp(useM,'1D'))
    mitigation_info = [mitigation_info, '_', int2str(num_MA_levels)];
elseif (strcmp(useM,'sweep'))
    mitigation_info = [mitigation_info, '_', int2str(num_MA_levels)];
elseif (strcmp(useM,'optimize'))
    mitigation_info = mitigation_info;
elseif (strcmp(useM,'random_search'))
    mitigation_info = [mitigation_info, '_', ...
        int2str(random_search_cases)];
elseif (strcmp(useM,'hybrid1D'))
    mitigation_info = [mitigation_info, '_', ...
        int2str(num_MA_levels), '_', int2str(random_search_cases)];
else
    mitigation_info = mitigation_info;
end

out_file = [out_folder, run_type, rel_info, sub_type, ...
    mitigation_info, '-', int2str(file_iteration), '.csv'];
while exist(out_file)
    file_iteration = file_iteration + 1;
    out_file = [out_folder, run_type, rel_info, sub_type, ...
        mitigation_info, '-', int2str(file_iteration), '.csv'];
end
```

```
function [met_constraints, case_constraints, BlockFuel_Des, ...
   Expected_BlockFuel_Des, Expected_BlockFuel_DOC, ...
   Expected_Range, Expected_TOFL, Expected_VAPP, ...
   Expected_RateofClimb, Expected_Span, Expected_RampWt, ...
   Expected_MaxFuel, Expected_OEW, Expected_Fuel_Used] = ...
   evaluate_constraints(Num_Constr, Num_Reliability_Cases, TWR, ...
   WSR, AR, EWMARG, FCDSUB, FACT, LapseRate, FPR, OPR, u_FCDSUB, ...
   u_EWMARG, u_AITEK, u_FACT,  MTOW_MA, OWE_delta, u_CLLDM, u_Rating)

if (Num_Reliability_Cases ~= size(u_FCDSUB,1))
    fprintf(1,'Num_Reliability_Cases (%i) ~= size(u_FCDSUB,1) ...
        (%i)!!!\n', Num_Reliability_Cases, size(u_FCDSUB,1));
end

constraints

met_constraints = min(min(case_constraints));

Expected_BlockFuel_Des = mean(BlockFuel_Des);
Expected_BlockFuel_DOC = mean(BlockFuel_DOC);
Expected_Range = mean(Range);
Expected_TOFL = mean(TOFL);
Expected_VAPP = mean(VAPP);
Expected_RateofClimb = mean(RateofClimb);
Expected_Span = mean(Span);
Expected_RampWt = mean(RampWt);
Expected_MaxFuel = mean(MaxFuel);
Expected_OEW = mean(OEW);
Expected_Fuel_Used = mean(Fuel_Used);
```

```
function [met_constraints, case_constraints, BlockFuel_DOC] = ...
    evaluate_constraints_MA(Num_Constr, Num_Reliability_Cases, ...
    TWR, WSR, AR, EWMARG, FCDSUB, FACT, LapseRate, FPR, OPR, ...
    u_FCDSUB, u_EWMARG, u_AITEK, u_FACT, ...
    MTOW_MA, OWE_delta, u_CLLDM, u_Rating, ...
    to_examine)

constraints

if to_examine > 0
    met_constraints = zeros(Num_Reliability_Cases, 1);
    for i = 1:Num_Constr
        met_constraints = met_constraints + ...
            max(10.^(-case_constraints(:, i)), 1);
    end
else
    met_constraints = transpose(min( transpose( case_constraints)));
end
```

```
function [met_constraints] = evaluate_constraints_MA_opt(MA_norm, ...
    Num_Constr, Num_Reliability_Cases, TWR, WSR, AR, EWMARG, ...
    FCDSUB, FACT, LapseRate, FPR, OPR, u_FCDSUB, u_EWMARG, ...
    u_AITEK, u_FACT, to_examine)

NN_Baselines_2014_08_19

MTOW_MA = MA_norm(1) * (MTOW_MA_max - MTOW_MA_min) + MTOW_MA_min;
u_CLLDM = MA_norm(2) * (u_CLLDM_max - u_CLLDM_min) + u_CLLDM_min;
u_Rating = MA_norm(3) * (u_Rating_max - u_Rating_min) + u_Rating_min;
calculate_MA_penalty;

constraints

if to_examine > 0
    met_constraints = zeros(Num_Reliability_Cases, 1);
    for i = 1:Num_Constr
        met_constraints = met_constraints + ...
            max(10.^(-case_constraints(:, i)), .999);
    end
    if MA_norm(1) < MTOW_MA_base
        MA_norm(1) = 0;
    end
    met_constraints = met_constraints + sum(MA_norm)/1000;

elseif to_examine == -100
    met_constraints = zeros(Num_Reliability_Cases, 1);

    for i = 1:Num_Constr
        met_constraints = met_constraints + ...
            max(10.^(-case_constraints(:, i)), .999);
    end
    if MA_norm(1) < MTOW_MA_base
        MA_norm(1) = 0;
    end
    met_constraints = met_constraints + sum(MA_norm)/1000;
else
    case_constraints
    min(transpose(case_constraints))
    met_constraints = max(min(transpose(case_constraints)))
end
```

```
function [desBlockFuel, Reliability, Reliability_MA, S_fail, ...
    S_fail_MA, Reliability_Individual, Expected_BlockFuel_Des, ...
    Expected_BlockFuel_DOC, Expected_Range, Expected_TOFL, ...
    Expected_VAPP, Expected_RateofClimb, Expected_Span, ...
    Expected_RampWt, Expected_MaxFuel, Expected_OEW, ...
    Expected_Fuel_Used] = Evaluate_X_S(x, S_Cases, ...
    Reliability_goal, Mitigation_Method, Num_Constr, ...
    num_MA_levels, random_search_cases, Mitigation_Weighting)

% Setup
```

```
NN_Baselines_2014_08_19
Num_Reliability_Cases = size(S_Cases,1);

% Reset Variables
Reliability = 0;
Reliability_MA = 0;
S_fail = S_Cases;
S_fail_MA = S_Cases;
Expected_BlockFuel_Des = 0;
Expected_BlockFuel_DOC = 0;
Reliability_Individual = zeros(Num_Constr,1);

% Start
Set_X;
MTOW_MA      = MTOW_MA_base;
OWE_delta    = OWE_delta_base;
u_CLLDM      = u_CLLDM_base;
u_Rating     = u_Rating_base;

[~, constraints_out, desBlockFuel, Expected_BlockFuel_Des, ...
    Expected_BlockFuel_DOC, Expected_Range, Expected_TOFL, ...
    Expected_VAPP, Expected_RateofClimb, Expected_Span, ...
    Expected_RampWt, Expected_MaxFuel, Expected_OEW, ...
    Expected_Fuel_Used] = evaluate_constraints(Num_Constr, 1, TWR, ...
    WSR, AR, EWMARG, FCDSUB, FACT, LapseRate, FPR, OPR, FCDSUB, ...
    EWMARG, (u_AITEK_max - u_AITEK_min)/2, FACT, MTOW_MA, ...
    OWE_delta, u_CLLDM, u_Rating);

if EWMARG > EWMARG_max || EWMARG < EWMARG_min
    Num_Reliability_Cases = 1;
    constraints_out(1,1) = -1;
    Expected_BlockFuel_DOC = Expected_BlockFuel_DOC * 2;
elseif FCDSUB > FCDSUB_max || FCDSUB < FCDSUB_min
    Num_Reliability_Cases = 1;
    constraints_out(1,1) = -1;
    Expected_BlockFuel_DOC = Expected_BlockFuel_DOC * 2;
elseif FACT > FACT_max || FACT < FACT_min
    Num_Reliability_Cases = 1;
    constraints_out(1,1) = -1;
    Expected_BlockFuel_DOC = Expected_BlockFuel_DOC * 2;
end

if Num_Reliability_Cases > 1 && ~(Reliability_goal == 0 && ...
    strcmp(Mitigation_Method,'none'))
    Set_S_matrix;

    [~, constraints_out, ~, Expected_BlockFuel_Des, ...
        Expected_BlockFuel_DOC, Expected_Range, Expected_TOFL, ...
        Expected_VAPP, Expected_RateofClimb, Expected_Span, ...
        Expected_RampWt, Expected_MaxFuel, Expected_OEW, ...
        Expected_Fuel_Used] = evaluate_constraints(Num_Constr, ...
        Num_Reliability_Cases, TWR, WSR, AR, EWMARG, FCDSUB, FACT, ...
        LapseRate, FPR, OPR, u_FCDSUB, u_EWMARG, u_AITEK, u_FACT, ...
        MTOW_MA, OWE_delta, u_CLLDM, u_Rating);
```

```
    constraints_passed = transpose(sum( transpose( ...
        constraints_out>0)));
    sum_success = sum(constraints_passed>=Num_Constr);
    mitigate = 1;

    % Catch cases which fail TOFL or Span.
    % - The impact of uncertain variables is incredibly small.
    % - Any failures means that the design should fail.
    min_constraints = min(constraints_out);
    if (min_constraints(1,2) < 0) || (min_constraints(1,5) < 0)
        sum_success = 0;
        if (min_constraints(1,5) < 0)
            % span is unrecoverable. Tell the mitigation solver ...
                not to bother
            mitigate = 0;
        end
    end
    Reliability = sum_success/Num_Reliability_Cases;
    if Reliability < Reliability_goal * 0.9;
        mitigate = 0;
    end

    if Mitigation_Weighting == 0
        mitigate = 0;
    end

    % initialize values
    Num_Mitigation_Cases = 1;
    sum_recovery = 0;

    if (Reliability > 1)
        fprintf(1,'Reliability = %f;  This should never ...
            happen!\n',Reliability);
    elseif(mitigate == 0)
        % something has told the algorithm not to mitigate
    elseif (Reliability == 1)
        %fprintf(1,'Full reliability\n');
    elseif (Reliability < 1)
        S_and_results = [S_Cases,  constraints_passed, ...
            constraints_out];
        S_and_results_sorted = sortrows(S_and_results, Num_S_Vars ...
            + 1);
        S_fail_and_results = ...
            S_and_results_sorted(1:(Num_Reliability_Cases - ...
            sum_success), :);
        S_fail_and_results(:, Num_S_Vars + 1) = rand(1, ...
            Num_Reliability_Cases - sum_success);
        S_fail_and_results = sortrows(S_fail_and_results, ...
            Num_S_Vars + 1);
        S_fail = S_fail_and_results(1:(Num_Reliability_Cases - ...
            sum_success), 1:Num_S_Vars);
        Num_Mitigation_Cases = size(S_fail, 1);

```

```
        %clear S_Cases
        S_Cases = S_fail;
        Set_S_matrix;

        if strcmp(Mitigation_Method,'none')

        elseif strcmp(Mitigation_Method,'sweep')
            constraints_passed_MA = zeros(Num_Mitigation_Cases, 1);

            if Num_Mitigation_Cases > 0
                for i = 0:num_MA_levels
                    for j = 0:num_MA_levels
                        for k = 0:num_MA_levels
                            MA_0 = [i, j, k] / num_MA_levels;
                            MA_0(1) = MA_0(1);

                            MTOW_MA  = MA_0(1) * (MTOW_MA_max  - ...
                                MTOW_MA_min)  + MTOW_MA_min;
                            u_CLLDM  = MA_0(2) * (u_CLLDM_max  - ...
                                u_CLLDM_min)  + u_CLLDM_min;
                            u_Rating = MA_0(3) * (u_Rating_max - ...
                                u_Rating_min) + u_Rating_min;
                            calculate_MA_penalty;
                            [~, constraints_out_MA, ~, ...
                                Expected_BlockFuel_Des, ...
                                Expected_BlockFuel_DOC] = ...
                                evaluate_constraints(Num_Constr, ...
                                Num_Mitigation_Cases, TWR, WSR, ...
                                AR, EWMARG, FCDSUB, FACT, ...
                                LapseRate, FPR, OPR, u_FCDSUB, ...
                                u_EWMARG, u_AITEK, u_FACT, ...
                                MTOW_MA, OWE_delta, u_CLLDM, ...
                                u_Rating);
                            constraints_passed_MA = ...
                                max(constraints_passed_MA, ...
                                transpose(sum( transpose( ...
                                constraints_out_MA > 0))));
                            sum_recovery = sum( ...
                                constraints_passed_MA >= Num_Constr);
                        end    % k
                    end    % j
                end    % i
            end

        elseif strcmp(Mitigation_Method,'optimize')

            % Don't waste time assessing low reliability cases
            assess_full_R_above = 0; %max(Reliability_goal, 0.75);
            minimum_assess_factor = 0.01;
            if ((Reliability < assess_full_R_above) && ...
                strcmp(Mitigation_Method, 'optimize'))
```

```
            Num_Mitigation_Cases = ...
                ceil(((1-assess_full_R_above - ...
                minimum_assess_factor) / assess_full_R_above^2 ...
                * Reliability^2 + minimum_assess_factor) * ...
                Num_Reliability_Cases);
        else
            Num_Mitigation_Cases = size(S_fail, 1);
        end
        S_fail_MA = ...
            S_fail_and_results(1:Num_Mitigation_Cases, ...
            1:Num_S_Vars);
        case_constraints_all = ...
            S_fail_and_results(1:Num_Mitigation_Cases, ...
            Num_S_Vars + 2:Num_S_Vars + 1 + Num_Constr);
        Num_Mitigation_Cases = size(S_fail_MA, 1);
        clear S_Cases
        S_Cases = S_fail_MA;
        Set_S_matrix;


        options = optimset('Display','off', 'Algorithm', 'sqp');
        MA_0   = [(MTOW_MA_base - MTOW_MA_min) / (MTOW_MA_max ...
            - MTOW_MA_min), (u_CLLDM_base - u_CLLDM_min) / ...
            (u_CLLDM_max - u_CLLDM_min), (u_Rating_base - ...
            u_Rating_min) / (u_Rating_max - u_Rating_min)];
        MA_LB = [0,  0,  0];        MA_UB = [1,  1,  1];
        MA_reset = MA_0;
        sqp_function_count = 0;
        MA_memory = zeros(Num_Mitigation_Cases, 3);

        if case_constraints_all(1,5) > 0    % if Span fails, ...
            it's unrecoverable. Don't waste time
            BF_DOC_keep = zeros(Num_Mitigation_Cases,1);
            for i = 1:Num_Mitigation_Cases
                clear S_Cases
                S_Cases = S_fail_MA(i,:);
                Set_S_matrix;
                [~, case_constraints_test] = ...
                    evaluate_constraints_MA(Num_Constr, 1, ...
                    TWR, WSR, AR, EWMARG, FCDSUB, FACT, ...
                    LapseRate, FPR, OPR, u_FCDSUB, u_EWMARG, ...
                    u_AITEK, u_FACT, .00001, OWE_delta_base, ...
                    u_CLLDM_base, u_Rating_base, 0);

                if (case_constraints_test(1,1) > 0) || ...
                    (case_constraints_test(1,6) > 0)
                    to_optimize    = @(MA_norm) ...
                        evaluate_constraints_MA_opt(MA_norm, ...
                        Num_Constr, 1, TWR, WSR, AR, EWMARG, ...
                        FCDSUB, FACT, LapseRate, FPR, OPR, ...
                        u_FCDSUB, u_EWMARG, u_AITEK, u_FACT, i);
                    MA_0 = MA_reset;

                    if case_constraints_all(i,1) < 0
```

```
                            MA_0(1) = 0.9;
                        end
                        if case_constraints_all(i,2) < 0 || ...
                            case_constraints_all(i,4) < 0
                            MA_0(3) = 0.5;
                        end
                        if case_constraints_all(i,3) < 0
                            MA_0(2) = 0.5;
                        end

                        [MA_out, FVAL, EXITFLAG, OUTPUT] = ...
                            fmincon(to_optimize, MA_0, [], [], [], ...
                            [], MA_LB, MA_UB, [], options);

                        MTOW_MA = MA_out(1) * (MTOW_MA_max - ...
                            MTOW_MA_min) + MTOW_MA_min;
                        u_CLLDM = MA_out(2) * (u_CLLDM_max - ...
                            u_CLLDM_min) + u_CLLDM_min;
                        u_Rating = MA_out(3) * (u_Rating_max - ...
                            u_Rating_min) + u_Rating_min;
                        calculate_MA_penalty;

                        [met_constraints, ~, BlockFuel_DOC_i] = ...
                            evaluate_constraints_MA(Num_Constr, 1, ...
                            TWR, WSR, AR, EWMARG, FCDSUB, FACT, ...
                            LapseRate, FPR, OPR, u_FCDSUB, ...
                            u_EWMARG, u_AITEK, u_FACT, MTOW_MA, ...
                            OWE_delta, u_CLLDM, u_Rating, 0);

                        BF_DOC_keep(i) = BlockFuel_DOC_i;
                        sqp_function_count = sqp_function_count + ...
                            OUTPUT.funcCount;

                        if met_constraints >= 0
                            sum_recovery = sum_recovery + 1;
                            MA_memory(i,:) = MA_out;
                        else
                            %fprintf(1,'Failed\n')
                        end
                    else
                        %fprintf(1,'Fuel Failed\n');
                    end
                end
            else
                %fprintf(1,'Span Failed\n');
            end
        elseif strcmp(Mitigation_Method,'random_search') || ...
            strcmp(Mitigation_Method,'1D') || ...
            strcmp(Mitigation_Method,'hybrid1D')

            M0 = [(MTOW_MA_base - MTOW_MA_min) / (MTOW_MA_max - ...
                MTOW_MA_min), (u_CLLDM_base - u_CLLDM_min) / ...
                (u_CLLDM_max - u_CLLDM_min), (u_Rating_base - ...
                u_Rating_min) / (u_Rating_max - u_Rating_min)];
```

```
            num_MA = size(M0,2);
            MA_LB = zeros(1,num_MA);
            MA_UB = ones(1,num_MA);

            recover_test = -1*ones(Num_Mitigation_Cases,1);
            M_keep   = ones(Num_Mitigation_Cases,1) * MA_UB * 999;
            M_scalar = ones(Num_Mitigation_Cases,1) * 999;
            BF_DOC_keep = zeros(Num_Mitigation_Cases,1);

            rand_Mitigation_array = rand(random_search_cases,num_MA);

            oneD_Mitigation_array = ones((num_MA_levels) * 3, 1) ...
                * M0;
            for i = 1:num_MA_levels
                for j = 1:size(M0,2)
                    if M0(j) == MA_LB(j)
                        oneD_Mitigation_array(i + ...
                            (j-1)*num_MA_levels, j) = i / ...
                            num_MA_levels;
                    elseif M0(j) == MA_UB(j)
                        oneD_Mitigation_array(i + ...
                            (j-1)*num_MA_levels, j) = (i-1) / ...
                            num_MA_levels;
                    else
                        if i/num_MA_levels < 0.5
                            oneD_Mitigation_array(i + (j-1) * ...
                                num_MA_levels, j) = (i-1) / ...
                                floor(num_MA_levels / 2) * (M0(j) ...
                                - MA_LB(j)) + MA_LB(j);
                        else
                            oneD_Mitigation_array(i + (j-1) * ...
                                num_MA_levels, j) = (i - ...
                                floor(num_MA_levels / 2)) / ...
                                ceil(num_MA_levels / 2) * ...
                                (MA_UB(j) - M0(j)) + M0(j);
                        end
                    end
                end
            end

            if (strcmp(Mitigation_Method, 'random_search'))
                Mitigation_array = rand_Mitigation_array;
            elseif (strcmp(Mitigation_Method, '1D'))
                Mitigation_array = oneD_Mitigation_array;
            elseif (strcmp(Mitigation_Method, 'hybrid1D'))
                Mitigation_array = [oneD_Mitigation_array; ...
                    rand_Mitigation_array];
            end

            for i = 1:size(Mitigation_array,1)
                mi = sumsqr(Mitigation_array(i,:) - M0);

                MTOW_MA  = Mitigation_array(i,1) .* (MTOW_MA_max ...
                    - MTOW_MA_min) + MTOW_MA_min;
```

```
                u_CLLDM   = Mitigation_array(i,2) .* (u_CLLDM_max ...
                    - u_CLLDM_min) + u_CLLDM_min;
                u_Rating = Mitigation_array(i,3) .* (u_Rating_max ...
                    - u_Rating_min) + u_Rating_min;
                calculate_MA_penalty;

                [met_constraints, ~, BlockFuel_DOC] = ...
                    evaluate_constraints_MA(Num_Constr, ...
                    Num_Mitigation_Cases, TWR, WSR, AR, EWMARG, ...
                    FCDSUB, FACT, LapseRate, FPR, OPR, u_FCDSUB, ...
                    u_EWMARG, u_AITEK, u_FACT, MTOW_MA, OWE_delta, ...
                    u_CLLDM, u_Rating, 0);

                M_scalar(met_constraints >= 0) = ...
                    min(M_scalar(met_constraints >= 0),mi);
                recover_test = max(recover_test,met_constraints);

                for j_ma = 1:num_MA
                    M_keep(M_scalar == mi, j_ma) = ...
                        Mitigation_array(i, j_ma);
                end
                BF_DOC_keep(M_scalar == mi) = ...
                    BlockFuel_DOC(M_scalar == mi);
            end
            sum_recovery = sum(recover_test >=0);
        else
            Mitigation_Method = 'not recognized';
            fprintf(1,'*** Unknown mitigation method specified! ...
                ***\n')
        end
    end

    Reliability_MA = sum_recovery / Num_Mitigation_Cases * (1 - ...
        Reliability) + Reliability;
    if exist('M_keep','var')
        M_keep = sortrows(M_keep);
        M_keep = M_keep(1:sum(M_keep(:,1)<999),:);
    end
else
    Reliability = 1;
    if min(constraints_out) < 0
        Reliability = 0;
    end
end

%Individual reliability
if exist('constraints_out','var')
    Reliability_Individual = ...
        transpose(sum(constraints_out>0)/Num_Reliability_Cases);
else
    Reliability_Individual = zeros(Num_Constr,1);
end

if exist('sum_recovery','var') && sum_recovery>0
```

```
    Recovered_BF_DOC = mean(BF_DOC_keep(BF_DOC_keep > 0));
    oldExpected_BlockFuel_DOC = Expected_BlockFuel_DOC;
    Expected_BlockFuel_DOC = (Expected_BlockFuel_DOC * ...
        Reliability + Recovered_BF_DOC * sum_recovery / ...
        Num_Mitigation_Cases * (1 - Reliability)) / Reliability_MA;
end
```

```
function [to_return, desBlockFuel, Reliability_MA, S_fail, ...
    S_fail_MA, Reliability] = fmincon_interface_EXS(x, S_Cases, ...
    Reliability_goal, Num_Constr, Mitigation_Weighting, ...
    Mitigation_Method, num_MA_levels, random_search_cases)

[desBlockFuel, Reliability, Reliability_MA, S_fail, S_fail_MA, ~, ...
    ~, Expected_BlockFuel_DOC] = Evaluate_X_S(x, S_Cases, ...
    Reliability_goal, Mitigation_Method, Num_Constr, ...
    num_MA_levels, random_search_cases, Mitigation_Weighting);
to_return = (1 - Mitigation_Weighting) *  ...
    Expected_BlockFuel_DOC/100000 - Mitigation_Weighting * ...
    Reliability_MA;
```

```
function [to_return, to_return2]= ...
    fmincon_interface_EXS_constraint(x, S_Cases, Reliability_goal, ...
    Success_goal, Num_Constr, Mitigation_Weighting, ...
    Mitigation_Method, num_MA_levels, random_search_cases)

if (Success_goal == 0)
    Mitigation_Weighting = 0;
    Mitigation_Method = 'none';
end

[~, ~, Reliability_MA, ~, ~, Reliability] = ...
    fmincon_interface_EXS(x, S_Cases, Reliability_goal, ...
    Num_Constr, 1, Mitigation_Method, num_MA_levels, ...
    random_search_cases);
to_return    = max(Reliability_goal - Reliability, Success_goal - ...
    Reliability_MA);
if Reliability_goal == 0 && Success_goal == 0
    if Reliability == 0
        to_return = 1;
    end
end
to_return2 = 0;
```

```
% NN_Baselines_2014_08_19.m

% ---- Design Variables ----
Num_X_Vars = 9;
% Baseline, min, and max values for NNs
TWR_base = 0.296195; TWR_min = 0.26;   TWR_max = 0.34;
```

```
WSR_base = 133.3391; WSR_min = 126;    WSR_max = 140;
AR_base = 10.0;      AR_min = 9.0;     AR_max = 11.0;
EWMARG_base = 0.035; EWMARG_min = 0.0; EWMARG_max = 0.06;
FCDSUB_base = 1.035; FCDSUB_min = 1.0; FCDSUB_max = 1.06;
FACT_base = 1.035;   FACT_min = 1.0;   FACT_max = 1.06;
LapseRate_base = 0.201439; LapseRate_min = 0.18; LapseRate_max = ...
    0.22;
FPR_base = 1.5;      FPR_min = 1.425;  FPR_max = 1.575;
OPR_base = 42.58;    OPR_min = 38.322; OPR_max = 46.838;


% ---- Uncertainty Variables ----
Num_S_Vars = 4;
u_FCDSUB_base = 1.0; u_FCDSUB_min = 0.99; u_FCDSUB_max = 1.06;
u_EWMARG_base = 0.0; u_EWMARG_min = -.01; u_EWMARG_max = 0.06;
u_AITEK_base = 1.95; u_AITEK_min = 1.9;   u_AITEK_max = 2.0;
u_FACT_base = 1.0;   u_FACT_min = 0.99;   u_FACT_max = 1.06;


% ---- Mitigation Variables ----
Num_M_Vars = 3;
MTOW_MA_base = 0.0;   MTOW_MA_min = -1000; MTOW_MA_max = 1900;
OWE_delta_base = 0.0; OWE_delta_min = 0.0; OWE_delta_max = 20000;
u_CLLDM_base = 2.413; u_CLLDM_min = 2.413; u_CLLDM_max = 2.53365;
u_Rating_base = 1.0;  u_Rating_min = 1.0;  u_Rating_max = 1.05;
```

```
% PrintResults.m

fprintf(1, '\n---------------------------\n');
fprintf(1, 'P(Compliance)        =    %5.3f,  Goal =    %5.3f\n', ...
    AA_final_Reliability, Reliability_goal);
fprintf(1, 'P(Success)           =    %5.3f,  Goal =    %5.3f\n', ...
    AA_final_Reliability_MA, Success_goal);
fprintf(1, 'E[BF_DOC]            = %8.1f,  Des  = %8.1f\n', ...
    Expected_BlockFuel_DOC, Expected_BlockFuel_Des);
fprintf(1, 'Mitigation_Weighting =    %5.3f\n', ...
    Mitigation_Weighting);
fprintf(1, '\n');

transpose_x = transpose(x);
Set_X;
AA_final_Output_Vector = transpose([TWR; WSR; AR; EWMARG; FCDSUB; ...
    FACT; LapseRate; FPR; OPR]);
```

```
function [S_Cases] = Random_Cases_Setup(Num_Reliability_Cases, ...
    Num_S_Vars, distribution_type)

% Generate random S settings
% Values will be from 0 to 1

if strcmp(distribution_type, 'normal')
```

```
    % Distribution is truncated at +/- 3 standard deviations
    std_deviations = 3;
    S_Cases = Truncated_Gaussian2(0, 1, Num_Reliability_Cases, ...
        Num_S_Vars, -std_deviations, std_deviations);
    S_Cases = S_Cases / (2*std_deviations) + .5;
end

if strcmp(distribution_type, 'Rayleigh')
    % Distribution is truncated at +3
    S_Cases = wblrnd(1, 2, Num_Reliability_Cases, Num_S_Vars);
    for i = 1:Num_Reliability_Cases
        for j = 1:Num_S_Vars
            if S_Cases(i,j) > 3
                iter_check = 0;
                while S_Cases(i,j) > 3 & iter_check < 5
                    S_Cases(i,j) = wblrnd(1,2,1,1);
                    iter_check = iter_check + 1;
                end
            end
        end
    end
    S_Cases = S_Cases / 3;
end

if strcmp(distribution_type, 'tri-centered')
    random1 = rand(Num_Reliability_Cases, Num_S_Vars);
    random2 = rand(Num_Reliability_Cases, Num_S_Vars);
    S_Cases = (random1 + random2) / 2;
end

if strcmp(distribution_type, 'uniform')
    S_Cases = rand(Num_Reliability_Cases,Num_S_Vars);
end

if strcmp(distribution_type, 'marg_set_distrib')
    if strcmp(version, '7.14.0.739 (R2012a)')
        % Other versions of MATLAB before R2014a will probably fail.
        fprintf(1, 'You set the distribution_type as ''%s''. ...
            \nThat won''t work in this version of MATLAB\n', ...
            distribution_type);
    else
        S_Cases = zeros(Num_Reliability_Cases, Num_S_Vars);
        b = [0.8396; 0.9303; 0.4812; 0.9206];
        S1_dist = makedist('Triangular', 'a', 0, 'b', b(1), 'c', 1);
        S2_dist = makedist('Triangular', 'a', 0, 'b', b(2), 'c', 1);
        S3_dist = makedist('Triangular', 'a', 0, 'b', b(3), 'c', 1);
        S4_dist = makedist('Triangular', 'a', 0, 'b', b(4), 'c', 1);
        S_Cases(:,1) = random(S1_dist, Num_Reliability_Cases, 1);
        S_Cases(:,2) = random(S2_dist, Num_Reliability_Cases, 1);
        S_Cases(:,3) = random(S3_dist, Num_Reliability_Cases, 1);
        S_Cases(:,4) = random(S4_dist, Num_Reliability_Cases, 1);
    end
end
```

```
%Set_S_matrix.m
% Set values for Uncertainty Variables based on standard normal ...
    inputs

%Uncertainty Variables
u_FCDSUB = S_Cases(:, 1) .* (u_FCDSUB_max - u_FCDSUB_min) + ...
    u_FCDSUB_min;
u_EWMARG = S_Cases(:, 2) .* (u_EWMARG_max - u_EWMARG_min) + ...
    u_EWMARG_min;
u_AITEK = S_Cases(:, 3) .* (u_AITEK_max - u_AITEK_min) + u_AITEK_min;
u_FACT = S_Cases(:, 4) .* (u_FACT_max - u_FACT_min) + u_FACT_min;
```

```
%Set_X.m
% Set Design Variable values

%Design Variables
TWR = x(1) * (TWR_max - TWR_min) + TWR_min;
WSR = x(2) * (WSR_max - WSR_min) + WSR_min;
AR = x(3) * (AR_max - AR_min) + AR_min;
EWMARG = x(4) * (EWMARG_max - EWMARG_min) + EWMARG_min;
FCDSUB = x(5) * (FCDSUB_max - FCDSUB_min) + FCDSUB_min;
FACT = x(6) * (FACT_max - FACT_min) + FACT_min;
LapseRate = x(i_set) * (LapseRate_max - LapseRate_min) + ...
    LapseRate_min;
FPR = x(i_set) * (FPR_max - FPR_min) + FPR_min;
OPR = x(i_set) * (OPR_max - OPR_min) + OPR_min;
```

```
% StartingPoint.m
NN_Baselines_2014_08_19;

% Design variables
TWR            = TWR_base;
WSR            = WSR_base;
AR             = AR_base;
LapseRate      = LapseRate_base;
FPR            = FPR_base;
OPR            = OPR_base;
% Margins
EWMARG         = EWMARG_base;
FCDSUB         = FCDSUB_base;
FACT           = FACT_base;

% Uncertain variables
u_FCDSUB       =   u_FCDSUB_base;
u_EWMARG       =   u_EWMARG_base;
u_AITEK        =   u_AITEK_base;
u_FACT         =   u_FACT_base;

% Mitigation variables
MTOW_MA        =   MTOW_MA_base;
```

```
OWE_delta   =   OWE_delta_base;
u_CLLDM     =   u_CLLDM_base;
u_Rating    =   u_Rating_base;


X0 = [TWR_base; WSR_base; AR_base; EWMARG_base; FCDSUB_base; ...
    FACT_base; LapseRate_base; FPR_base; OPR_base];
LB_scaled = [TWR_min; WSR_min; AR_min; EWMARG_min; FCDSUB_min; ...
    FACT_min; LapseRate_min; FPR_min; OPR_min];
UB_scaled = [TWR_max; WSR_max; AR_max; EWMARG_max; FCDSUB_max; ...
    FACT_max; LapseRate_max; FPR_max; OPR_max];

X0 = (X0 - LB_scaled) ./ (UB_scaled - LB_scaled);
LB = zeros(size(X0,1),1);
UB = ones(size(X0,1),1);
```

```
%unscale_x.m
% Change scaled x values (from input DoE) to unscaled values for ...
    optimizer

x(1) = (TWR - TWR_min) / (TWR_max - TWR_min);
x(2) = (WSR - WSR_min) / (WSR_max - WSR_min);
x(3) = (AR - AR_min) / (AR_max - AR_min);
x(4) = (EWMARG - EWMARG_min) / (EWMARG_max - EWMARG_min);
x(5) = (FCDSUB - FCDSUB_min) / (FCDSUB_max - FCDSUB_min);
x(8) = (FACT - FACT_min) / (FACT_max - FACT_min);
x(7) = (LapseRate - LapseRate_min) / (LapseRate_max - LapseRate_min);
x(8) = (FPR - FPR_min) / (FPR_max - FPR_min);
x(9) = (OPR - OPR_min) / (OPR_max - OPR_min);
```

```
% WriteOutput_case.m

fid = fopen(out_file, 'a+');

% Results
for i_out = 1:size(x_array, 1)
    fprintf(fid, '%f, ', x_array(i_out, i_main));
end

% Scaled X values
fprintf(fid, '%f, ', x_array(1, i_main) * (TWR_max - TWR_min) + ...
    TWR_min);
fprintf(fid, '%f, ', x_array(2, i_main) * (WSR_max - WSR_min) + ...
    WSR_min);
fprintf(fid, '%f, ', x_array(3, i_main) * (AR_max - AR_min) + ...
    AR_min);
fprintf(fid, '%f, ', x_array(4, i_main) * (EWMARG_max - ...
    EWMARG_min) + EWMARG_min);
fprintf(fid, '%f, ', x_array(5, i_main) * (FCDSUB_max - ...
    FCDSUB_min) + FCDSUB_min);
```

```
fprintf(fid, '%f, ', x_array(6, i_main) * (FACT_max - FACT_min) + ...
    FACT_min);
fprintf(fid, '%f, ', x_array(7, i_main) * (LapseRate_max - ...
    LapseRate_min) + LapseRate_min);
fprintf(fid, '%f, ', x_array(8, i_main) * (FPR_max - FPR_min) + ...
    FPR_min);
fprintf(fid, '%f, ', x_array(9, i_main) * (OPR_max - OPR_min) + ...
    OPR_min);

fprintf(fid, '%f, ', AA_array_Reliability(i_main));
fprintf(fid, '%f, ', AA_array_Reliability_MA(i_main));
fprintf(fid, '%f, ', AA_array_Block_Fuel(i_main));
fprintf(fid, '%f, ', AA_array_Expected_BF_Des(i_main));
fprintf(fid, '%f, ', AA_array_Expected_BF_DOC(i_main));
fprintf(fid, '%f, ', AA_array_Reliability_Individual(:, i_main));
fprintf(fid, '%f, ', function_calls_array(i_main));
fprintf(fid, '%f, ', Mitigation_Weight_out_array(i_main));
fprintf(fid, '%f, ', Reliability_goal_out_array(i_main));
fprintf(fid, '%f, ', Success_goal_out_array(i_main));


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Global Optimizer Results
for i_out = 1:size(xatmin_array, 1)
     fprintf(fid, '%f, ', xatmin_array(i_out, i_main));
end

fprintf(fid, '%f, ', xatmin_array(1, i_main) * (TWR_max - ...
    TWR_min) + TWR_min);
fprintf(fid, '%f, ', xatmin_array(2, i_main) * (WSR_max - ...
    WSR_min) + WSR_min);
fprintf(fid, '%f, ', xatmin_array(3, i_main) * (AR_max - AR_min) ...
    + AR_min);
fprintf(fid, '%f, ', xatmin_array(4, i_main) * (EWMARG_max - ...
    EWMARG_min) + EWMARG_min);
fprintf(fid, '%f, ', xatmin_array(5, i_main) * (FCDSUB_max - ...
    FCDSUB_min) + FCDSUB_min);
fprintf(fid, '%f, ', xatmin_array(6, i_main) * (FACT_max - ...
    FACT_min) + FACT_min);
fprintf(fid, '%f, ', xatmin_array(7, i_main) * (LapseRate_max - ...
    LapseRate_min) + LapseRate_min);
fprintf(fid, '%f, ', xatmin_array(8, i_main) * (FPR_max - ...
    FPR_min) + FPR_min);
fprintf(fid, '%f, ', xatmin_array(9, i_main) * (OPR_max - ...
    OPR_min) + OPR_min);

fprintf(fid, '%f, ', AA_Global_Reliability(i_main));
fprintf(fid, '%f, ', AA_Global_Reliability_MA(i_main));
fprintf(fid, '%f, ', AA_Global_Block_Fuel(i_main));
fprintf(fid, '%f, ', AA_Global_Expected_BF_Des(i_main));
fprintf(fid, '%f, ', AA_Global_Expected_BF_DOC(i_main));
fprintf(fid, '%f, ', AA_Global_Reliability_Individual(:, i_main));
fprintf(fid, '%f, ', function_calls_global_array(i_main));
```

```
if exist('Expected_Range', 'var')
    fprintf(fid, '%f, ', Expected_Range);
    fprintf(fid, '%f, ', Expected_TOFL);
    fprintf(fid, '%f, ', Expected_VAPP);
    fprintf(fid, '%f, ', Expected_RateofClimb);
    fprintf(fid, '%f, ', Expected_Span);
    fprintf(fid, '%f, ', Expected_RampWt);
    fprintf(fid, '%f, ', Expected_MaxFuel);
    fprintf(fid, '%f, ', Expected_OEW);
    fprintf(fid, '%f, ', Expected_Fuel_Used);
end

fprintf(fid, '\n');
fclose(fid);
```

```
% WriteOutput_setup.m

fprintf(1, 'Setting up output file: %s\n', out_file);
fid = fopen(out_file,'w');

% Basic information
fprintf(fid, 'Settings\n');
fprintf(fid, 'Num_Reliability_Cases,%i\n', Num_Reliability_Cases);
fprintf(fid, 'optimization_method,%s\n', optimization_method);
if strcmp(optimization_method, 'DIRect')
    fprintf(fid, 'DIRect_maxevals,%i\n', DIRect_maxevals);
elseif strcmp(optimization_method, 'GA')
    fprintf(fid, 'GA_PopulationSize,%i\n', GA_PopulationSize);
end
fprintf(fid, 'distribution_type,%s\n', distribution_type);
fprintf(fid, 'Mitigation_Method,%s\n', Mitigation_Method);
useM = Mitigation_Method;
if strcmp(Mitigation_Method, 'none')
    useM = default_Mitigation_Method;
    fprintf(fid, 'default_Mitigation_Method,%s\n', ...
        default_Mitigation_Method);
end

if (strcmp(useM,'1D'))
    fprintf(fid, 'num_MA_levels,%i\n',num_MA_levels);
elseif (strcmp(useM,'sweep'))
    fprintf(fid, 'num_MA_levels,%i\n',num_MA_levels);
elseif (strcmp(useM,'optimize'))

elseif (strcmp(useM,'random_search'))
    fprintf(fid, 'random_search_cases,%i\n',random_search_cases);
elseif (strcmp(useM,'hybrid1D'))
    fprintf(fid, 'num_MA_levels,%i\n',num_MA_levels);
    fprintf(fid, 'random_search_cases,%i\n',random_search_cases);
else

end
```

```

if strcmp(run_type, 'DOE_')
    fprintf(fid, 'doe_filename,%s\n', doe_filename);
    fprintf(fid, 'x_cases,%i\n', x_cases);
elseif strcmp(run_type, 'MCS_')
    fprintf(fid, 'x_cases,%i\n', x_cases);
elseif strcmp(run_type, 'Opt_')
    if size(Reliability_Goal_Array, 2) > 1
        fprintf(fid, 'size(Reliability_Goal_Array,2),%i\n', ...
            size(Reliability_Goal_Array, 2));
    else
        fprintf(fid, 'Reliability_Goal,%f\n', ...
            Reliability_Goal_Array(1));
    end
    if size(Success_Goal_Array,2) > 1
        fprintf(fid, 'size(Success_Goal_Array,2),%i\n', ...
            size(Success_Goal_Array, 2));
    else
        fprintf(fid, 'Success_Goal,%f\n',S uccess_Goal_Array(1));
    end
end
fprintf(fid, '\n');

% Results
%fprintf(fid, 'Optimizer Reults\n');
% X's
for i = 1:size(x_array,1)
    fprintf(fid, 'x%s,',int2str(i));
end

% Scaled X values
fprintf(fid, 'TWR, WSR, AR, h_EW, h_Drag, h_FuelFlow, LapseRate, ...
    FPR, OPR,');

% Responses
fprintf(fid, 'Compliance, Success, BF_Det, E[BF_Des], E[BF_DOC],');
for i = 1:size(AA_array_Reliability_Individual, 1)
    if i == 1
        fprintf(fid, 'Comp_Range,');
    elseif i == 2
        fprintf(fid, 'Comp_TOFL,');
    elseif i == 3
        fprintf(fid, 'Comp_Vapp,');
    elseif i == 4
        fprintf(fid, 'Comp_RoC,');
    elseif i == 5
        fprintf(fid, 'Comp_Span,');
    elseif i == 6
        fprintf(fid, 'Comp_Fuel,');
    else
        fprintf(fid, 'Compliance%s,', int2str(i));
    end
end
```

```
fprintf(fid, 'SQP Function Calls, Mitigation_Wt, ...
    Reliability_goal, Success_goal,');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Global Optimizer Results
for i = 1:size(xatmin_array,1)
    fprintf(fid, 'x_global%s,', int2str(i));
end

fprintf(fid, 'TWR_global, WSR_global, AR_global, h_EW_global, ...
    h_Drag_global, h_FuelFlow_global, LapseRate_global, ...
    FPR_global, OPR_global, Global_Compliance, Global_Success, ...
    Global_BF_DES_Det, Global_E[BF_Des], Global_E[BF_DOC],');
for i = 1:size(AA_Global_Reliability_Individual, 1)
    if i == 1
        fprintf(fid, 'G_Comp_Range,');
    elseif i == 2
        fprintf(fid, 'G_Comp_TOFL,');
    elseif i == 3
        fprintf(fid, 'G_Comp_Vapp,');
    elseif i == 4
        fprintf(fid, 'G_Comp_RoC,');
    elseif i == 5
        fprintf(fid, 'G_Comp_Span,');
    elseif i == 6
        fprintf(fid, 'G_Comp_Fuel,');
    else
        fprintf(fid, 'Global_P(Comp)_Ind_%s,', int2str(i));
    end
end
fprintf(fid, 'Global Function Calls,');

fprintf(fid, 'E[Range], E[TOFL], E[VAPP], E[RateofClimb], ...
    E[Span], E[RampWt], E[MaxFuel], E[OEW], E[Fuel_Used],');

fprintf(fid, '\n');
fclose(fid);
```

REFERENCES

[1] Allaix, D. and Carbone, V., "An improvement of the response surface method," *Structural Safety*, vol. 33, no. 2, pp. 165 – 172, 2011.

[2] Anderson, J. D., *Aircraft Performance and Design.* Boston, MA: WCB / McGraw-Hill, 1999.

[3] Bae, H.-R., Grandhi, R. V., and Canfield, R. A., "Reliability-based design optimization under imprecise uncertainty," in *Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Material Conference*, vol. 2069, (Austin, TX), American Institute of Aeronautics and Astronautics, 2005.

[4] Bandte, O., *A probabilistic multi-criteria decision making technique for conceptual and preliminary aerospace systems design.* PhD thesis, Georgia Institute of Technology, 2000.

[5] Barbarosoglu, G. and Arda, Y., "A two-stage stochastic programming framework for transportation planning in disaster response," *Journal of the Operational Research Society*, vol. 55, no. 1, pp. 43–53, 2004.

[6] Beale, E. M., "On minimizing a convex function subject to linear inequalities," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 173–184, 1955.

[7] Beyer, H.-G. and Sendhoff, B., "Robust optimization - a comprehensive survey," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 33-34, pp. 3190 – 3218, 2007.

[8] Birge, J. R. and Louveaux, F., "Introduction to stochastic programming," *Interfaces -Providence- Institute of Management Sciences*, vol. 28, no. 6, p. 79, 1998.

[9] Bishop, C. M. and others, *Neural networks for pattern recognition.* Clarendon press Oxford, 1995.

[10] Boeing Commercial Airplanes, Seattle, WA, *777-200/300 Airplane Characteristics for Airport Planning*, d6-58329 ed., May 2011.

[11] Bordley, R. F. and Pollock, S. M., "A decision-analytic approach to reliability-based design optimization," *Operations research*, vol. 57, no. 5, pp. 1262–1270, 2009.

[12] Borer, N. K., *Decision Making Strategies for Probabilistic Aerospace Systems Design.* PhD thesis, Georgia Institute of Technology, 2006.

[13] Box, G. E. and Draper, N. R., *Introduction to response surface methodology.* Wiley Online Library, 1987.

[14] Briceo, S. I., *A game-based decision support methodology for competitive systems design.* PhD thesis, Georgia Institute of Technology, 2008.

[15] Bristow, J. W. and Irving, P., "Safety factors in civil aircraft design requirements," *Engineering Failure Analysis*, vol. 14, no. 3, pp. 459 – 470, 2007.

[16] Bucher, C. and Macke, M., "Response surfaces for reliability assessment," in *Engineering Design Reliability Handbook* (Nikolaidis, E., Ghiocel, D. M., and Singhal, S., eds.), CRC, 2005.

[17] Bucher, C. and Most, T., "A comparison of approximate response functions in structural reliability analysis," *Probabilistic Engineering Mechanics*, vol. 23, no. 2, pp. 154–163, 2008.

[18] Chen, X., Hasselman, T. K., and Neill, D. J., "Reliability based structural design optimization for practical applications," in *Proceedings of the 38th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, pp. 2724–2732, 1997.

[19] Choi, B.-L., Choi, J.-H., and Choi, D.-H., "Reliability-based design optimization using enhanced initial design and two-point approximation technique," in *10 th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004.

[20] Choi, T. P., *A Recourse-Based Solution Approach to the Design of Fuel Cell Aeropropulsion Systems.* PhD thesis, Georgia Institute of Technology, 2008.

[21] Daberkow, D. D. and Mavris, D. N., "New approaches to conceptual and preliminary aircraft design: A comparative assessment of a neural network formulation and a response surface methodology," in *1998 World Aviation Conference*, American Institute of Aeronautics and Astronautics, September 1998.

[22] Daly, M. and Gunston, B., *Janes aero-engines.* McGraw-Hill, 2007.

[23] Dantzig, G. B., "Linear programming under uncertainty," *Management science*, vol. 1, no. 3-4, pp. 197–206, 1955.

[24] Daskilewicz, M. J., German, B. J., Takahashi, T. T., Donovan, S., and Shajanian, A., "Effects of disciplinary uncertainty on multi-objective optimization in aircraft conceptual design," *Structural and Multidisciplinary Optimization*, vol. 44, no. 6, pp. 831–846, 2011.

[25] Du, X., "First order and second reliability methods," in *Probabilistic Engineering Design*, Rolla, MO: University of Missouri - Rolla, 2005.

[26] Du, X. and Chen, W., "Sequential optimization and reliability assessment method for efficient probabilistic design," *Journal of Mechanical Design*, vol. 126, pp. 225 – 233, 2004.

[27] Enevoldsen, I. and Srensen, J. D., "Reliability-based optimization of series systems of parallel systems," *Journal of Structural Engineering*, vol. 119, pp. 1069 – 1084, 1993.

[28] Feldman, A. M., "Bilateral trading processes, pairwise optimality, and pareto optimality," *The Review of Economic Studies*, vol. 40, no. 4, pp. pp. 463–473, 1973.

[29] Filippone, A., "Comprehensive analysis of transport aircraft flight performance," *Progress in Aerospace Sciences*, vol. 44, no. 3, pp. 192 – 236, 2008.

[30] Frangopol, D. M. and Maute, K., "Life-cycle reliability-based optimization of civil and aerospace structures," *Computers & Structures*, vol. 81, no. 7, pp. 397 – 410, 2003.

[31] Fujita, M. and Rackwitz, R., "Updating first-and second-order reliability estimates by importance sampling.," *Structural Eng. / Earthquake Eng.*, no. 392, pp. 53–59, 1988.

[32] Haldar, A. and Mahadevan, S., *Probability, Reliability, and Statistical Methods in Engineering Design.* New York: John Wiley & Sons, Inc, 2000.

[33] Hasofer, A. and Lind, N., "An exact and invariant first-order reliability format," *Journal of the Engineering Mechanics Division*, vol. 100, no. EM1, pp. 111–121, 1974.

[34] Haugh, M., "The monte carlo framework, examples from finance and generating correlated random variables," *Course Notes*, 2004.

[35] Helton, J. C. and Burmaster, D. E., "Guest editorial: treatment of aleatory and epistemic uncertainty in performance assessments for complex systems," *Reliability Engineering & System Safety*, vol. 54, no. 2, pp. 91–94, 1996.

[36] Helton, J. C., Johnson, J. D., Sallaberry, C. J., and Storlie, C. B., "Survey of sampling-based methods for uncertainty and sensitivity analysis," *Reliability Engineering & System Safety*, vol. 91, no. 10, pp. 1175–1209, 2006.

[37] Hill, W. J. and Hunter, W. G., "A review of response surface methodology: A literature survey," *Technometrics*, vol. 8, no. 4, pp. pp. 571–590, 1966.

[38] Hui, F. and Weiji, L., "An efficient method for reliability-based multidisciplinary design optimization," *Chinese Journal of Aeronautics*, vol. 21, no. 4, pp. 335 – 340, 2008.

[39] Hurtado, J. and Barbat, A., "Monte carlo techniques in computational stochastic mechanics," *Archives of Computational Methods in Engineering*, vol. 5, no. 1, pp. 3–29, 1998.

[40] Huyse, L., "Solving problems of optimization under uncertainty as statistical decision problems," *AIAA paper*, vol. 1519, 2001.

[41] Jane, F. T. and others, *Jane's all the world's aircraft*. McGraw-Hill, 2014.

[42] Johnson, C. and Schutte, J., "Basic regression analysis for integrated neural networks (brainn) documentation," *Masters Project, Georgia Institute of Technology, Atlanta, GA, June*, 2007.

[43] Kandebo, S. W., "Putting f135 to the test," *Aviation Week & Space Technology*, vol. 160, no. 12, pp. 52 – 54, 2004.

[44] Keane, A. and Nair, P., *Computational Approaches for Aerospace Design: the Pursuit of Excellence*. Chichester, England; Hoboken, N.J.: John Wiley & Sons, 2005.

[45] Kirby, M. R., *A methodology for technology identification, evaluation, and selection in conceptual and preliminary aircraft design*. PhD thesis, Georgia Institute of Technology, 2001.

[46] Lian, Y. and Nam-Ho, K., "Reliability-based design optimization of a transonic compressor.," *AIAA Journal*, vol. 44, no. 2, pp. 368 – 375, 2006.

[47] Lim, D., *A systematic approach to design for lifelong aircraft evolution*. PhD thesis, Georgia Institute of Technology, 2009.

[48] Liu, H., Chen, W., and Sudjianto, A., "Relative entropy based method for probabilistic sensitivity analysis in engineering design," *Journal of Mechanical Design*, vol. 128, p. 326, 2006.

[49] Liu, P.-L. and Der Kiureghian, A., "Optimization algorithms for structural reliability," *Structural safety*, vol. 9, no. 3, pp. 161–177, 1991.

[50] Liu, Y. W. and Moses, F., "A sequential response surface method and its application in the reliability analysis of aircraft structural systems," *Structural Safety*, vol. 16, no. 1, pp. 39–46, 1994.

[51] Madsen, H. O., Krenk, S., and Lind, N. C., *Methods of Structural Safety*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[52] Mahadevan, S., *Stochastic finite element-based structural reliability analysis and optimization*. PhD thesis, Georgia Institute of Technology, 1988.

[53] Majumder, L. and Rao, S., "Interval-based optimization of aircraft wings under landing loads," *Computers & Structures*, vol. 87, no. 34, pp. 225 – 235, 2009.

[54] Martin, J. D., *A Methodology for Evaluating System-Level Uncertainty.* PhD thesis, The Pennsylvania State University, 2005.

[55] Marx, W. J., Mavris, D. N., and Schrage, D. P., "Integrated product development for the wing structural design of the high speed civil transport," in *the 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City, FL, September*, pp. 7–9, 1994.

[56] Mavris, D. N., Bandte, O., and DeLaurentis, D. A., "Robust design simulation: a probabilistic approach to multidisciplinary design," *Journal of Aircraft*, vol. 36, no. 1, pp. 298–307, 1999.

[57] Mavris, D. N. and DeLaurentis, D. A., "A probabilistic approach for examining aircraft concept feasibility and viability," *Aircraft Design*, vol. 3, no. 2, pp. 79 – 101, 2000.

[58] Mavris, D. and Birney, M., "Formulation of a stochastic, physics-based strategic business decision making environment," *AIAA Paper*, vol. 5850, pp. 1–3, 2002.

[59] McCullers, L., *FLOPS Flight Optimization System Release 8.11 User's Guide.* NASA Langley Research Center, Hampton, VA, October 2009.

[60] McDonald, M. and Mahadevan, S., "Design optimization with system-level reliability constraints," *Journal of Mechanical Design*, vol. 130, p. 021403, 2008.

[61] Myers, R. H., Khuri, A. I., and Carter, W. H., "Response surface methodology: 1966–l988," *Technometrics*, vol. 31, pp. 137–157, September 1989.

[62] Nam, T., *A Generalized Sizing Method for Revolutionary Concepts under Probabilistic Design Constraints.* PhD thesis, Georgia Institute of Technology, 2007.

[63] Nam, T. and Mavris, D. N., "Multi-stage reliability-based design optimization for aerospace system conceptual design," in *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 16th AIAA/ASME/AHS Adaptive Structures Conference,10th AIAA Non-Deterministic Approaches Conference, 9th AIAA Gossamer Spacecraft Forum, 4th AIAA Multidisciplinary Design Optimization Specialists Conference*, vol. 2073, April 2008.

[64] Nam, T. and Mavris, D. N., "Multi-stage reliability-based design optimization for aerospace system conceptual design." Unpublished update to previous work submitted to American Institute of Aeronautics and Astronautics, 2012.

[65] Nam, T., Soban, D. S., and Mavris, D. N., "A non-deterministic aircraft sizing method under probabilistic design constraints," in *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, vol. 2062, 2006.

[66] National Aeronautics and Space Administration, Cleveland, OH, *NPSS User's Guide*, npss_1.6.4, rev: q. ed., 2006.

[67] Norris, G., "P&w plans for hybrid pw4000," *Flight International*, vol. 10, p. 4, December 1997.

[68] Norris, G. and Wagner, M., "Boeing 777. st. paul, minnesota: Motorbooks international," tech. rep., ISBN 0-7603-0091-7, 1996.

[69] Oberkampf, W. L., Helton, J. C., Joslyn, C. A., Wojtkiewicz, S. F., and Ferson, S., "Challenge problems: uncertainty in system response given uncertain parameters," *Reliability Engineering & System Safety*, vol. 85, no. 13, pp. 11 – 19, 2004.

[70] Ordaz, I., *A probabilistic and multi-objective conceptual design methodology for the evaluation of thermal management systems on air-breathing hypersonic vehicles.* PhD thesis, Georgia Institute of Technology, 2008.

[71] Padmanabhan, D., *Reliability-based Optimization for Multidisciplinary System Design.* PhD thesis, University of Notre Dame, 2003.

[72] Padmanabhan, D., Tappeta, R. V., and Batill, S., "Monte carlo simulation in reliability based optimization applied to multidisciplinary system design," in *Proceedings of the 44th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference*, pp. 2003–1503, 2003.

[73] Paiva, R. M., Crawford, C., and Suleman, A., "Robust and reliability based design optimization framework for wing design," in *2nd International Conference on Engineering Optimization*, vol. 1189, (Lisbon, Portugal), september 2010.

[74] Rackwitz, R. and Flessler, B., "Structural reliability under combined random load sequences," *Computers & Structures*, vol. 9, no. 5, pp. 489–494, 1978.

[75] Raymer, D. P., *Aircraft Design: a Conceptual Approach*, vol. 3. Reston, VA: American Institute of Aeronautics and Astronautics, 1999.

[76] Robinson, D. G., "A survey of probabilistic methods used in reliability, risk and uncertainty analysis: Analytical techniques i," *Sandia National Lab, Report SAND98*, vol. 1189, 1998.

[77] Rosenblatt, M., "Remarks on a multivariate transformation," *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. pp. 470–472, 1952.

[78] Roskam, J., *Airplane Design Part I: Preliminary Sizing of Airplanes.* Lawrence, KS: DARcorporation, third ed., 2003.

[79] Sahinidis, N. V., "Optimization under uncertainty: state-of-the-art and opportunities," *Computers & Chemical Engineering*, vol. 28, no. 67, pp. 971 – 983, 2004.

[80] Samuelson, P. A., "The pure theory of public expenditure," *The Review of Economics and Statistics*, vol. 36, no. 4, pp. pp. 387–389, 1954.

[81] Schalkoff, R. J., *Artificial neural networks.* McGraw-Hill New York, 1997.

[82] Schrage, D. P., "Technology for rotorcraft affordability through integrated product/process development (ippd)," in *Proceedings of the 55th National Forum of the American Helicopter Society*, (Montreal, Quebec, Canada), Georgia Institute of Technology, May 1999.

[83] Shah, J. and Hazelrigg, H., "Research opportunities in engineering design," in *NSF Strategic Planning Workshop, Final Report*, 1996.

[84] Smith, N. and Mahadevan, S., "Probabilistic methods for aerospace system conceptual design.," *Journal of Spacecraft and Rockets*, vol. 40, no. 3, pp. 411–418, 2003.

[85] Soban, D. S. and Mavris, D. N., "Assessing the impact of technology on aircraft systems using technology impact forecasting," *Journal of Aircraft*, p. 14, September 2013.

[86] Sutter, J. and Spenser, J., *747: Creating the World's First Jumbo Jet and Other Adventures from a Life in Aviation.* HarperCollins, 2010.

[87] Taguchi, G., *Introduction to Quality Engineering: Designing Quality into Products and Processes.* Dearborn, MI: Asian Productivity Organization, 1986.

[88] Tong, M. T. and Naylor, B. A., "An object-oriented computer code for aircraft engine weight estimation," in *ASME Turbo Expo 2008: Power for Land, Sea, and Air*, pp. 1–7, American Society of Mechanical Engineers, 2008.

[89] Torng, T. and Yang, R., "Advanced reliability based optimization method for robust structural system design.," in *Collection of Technical Papers - AIAA / ASME Structures, Structural Dynamics and Materials Conference*, (Southwest Research Inst), pp. 1198–1206, 1993.

[90] Tu, J., Choi, K. K., and Park, Y. H., "A new study on reliability-based design optimization," *Journal of Mechanical Design*, vol. 121, pp. 557–564, 1999.

[91] Walsh, P. P. and Fletcher, P., *Gas Turbine Performance.* Oxford, UK: Blackwell Science Ltd, second ed., 2004.

[92] Wang, L., Grandhi, R. V., and Hopkins, D. A., "Structural reliability optimization using an efficient safety index calculation procedure.," *International Journal for Numerical Methods in Engineering*, vol. 38, no. 10, p. 1721, 1995.

[93] Yang, J., Huang, H.-Z., He, L.-P., Zhu, S.-P., and Wen, D., "Risk evaluation in failure mode and effects analysis of aircraft turbine rotor blades using dempstershafer evidence theory under uncertainty," *Engineering Failure Analysis*, vol. 18, no. 8, pp. 2084 – 2092, 2011.

[94] Yang, R., Choi, K., Gu, L., Tho, C., and Youn, B., "Reliability-based multidisciplinary design optimization of a full vehicle system.," in *Papers - American Institute of Aeronautics and Astronautics*, vol. 6 of *Structures, structural dynamics, and materials conference; AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference 10th AIAA/ASME/ASCE/AHS adaptive structures forum, 4th AIAA non-deterministic approaches forum, 3rd AIAA gossamer spacecraft forum*, pp. 4174 – 4182, 2002.

[95] Yao, W., Chen, X., Luo, W., van Tooren, M., and Guo, J., "Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles," *Progress in Aerospace Sciences*, vol. 47, no. 6, pp. 450–479, 2011.

[96] Zang, T. A., Hemsch, M. J., Hilburger, M. W., Kenny, S. P., Luckring, J. M., Maghami, P., Padula, S. L., and Stroud, W. J., *Needs and Opportunities for Uncertainty-Based Multidisciplinary Design Methods for Aerospace Vehicles.* National Aeronautics and Space Administration, Langley Research Center, 2002.

[97] Zhao, Y.-G. and Ono, T., "A general procedure for first/second-order reliabilitymethod (form/sorm)," *Structural Safety*, vol. 21, no. 2, pp. 95–112, 1999.

[98] Zhu, T.-L., "A reliability-based safety factor for aircraft composite structures," *Computers & Structures*, vol. 48, no. 4, pp. 745 – 748, 1993.

[99] Zou, T. and Mahadevan, S., "Multiobjective reliability-based optimization.," in *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, vol. 6, (Vanderbilt University), pp. 4105–4111, 2004.

[100] Zou, T. and Mahadevan, S., "Versatile formulation for multiobjective reliability-based design optimization," *Journal of Mechanical Design*, vol. 128, no. 6, pp. 1217–1226, 2006.

VITA

Joseph Scott Wilson was born in Miami, Florida in 1981. He earned his Bachelor of Science degree in Aerospace Engineering from the Georgia Institute of Technology in 2005. He then continued his post-graduate education at the Georgia Tech, receiving his Master of Science degree in Aerospace Engineering in 2007. Will complete his Ph.D. in Aerospace Engineering in May of 2015.

During his undergradute and masters time at Tech, J. Scott spent much of his non-school time playing the sousaphone in the Georgia Tech Marching Band and Georgia Tech Pep Band.