Multidisciplinary Methods for Performing Trade Studies on Blended Wing Body Aircraft

by

Cory Asher Kays

S.B., Massachusetts Institute of Technology (2011)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aerospace Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

Author .....
Department of Aeronautics and Astronautics
May 20, 2013

Certified by .....
Karen E. Willcox
Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by .....
Eytan H. Modiano
Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Multidisciplinary Methods for Performing Trade Studies on Blended Wing Body Aircraft

by

Cory Asher Kays

Submitted to the Department of Aeronautics and Astronautics
on May 20, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science in Aerospace Engineering

Abstract

Multidisciplinary design optimization (MDO) is becoming an essential tool for the design of engineering systems due to the inherent coupling between discipline analyses and the increasing complexity of such systems. An important component of MDO is effective exploration of the design space since this is often a key driver in finding characteristics of systems which perform well. However, many design space exploration techniques scale poorly with the number of design variables and, moreover, a large-dimensional design space can be prohibitive to designer manipulation.

This research addresses complexity management in trade-space exploration of multidisciplinary systems, with a focus on the conceptual design of Blended Wing Body (BWB) aircraft. The objectives of this thesis are twofold. The first objective is to create a multidisciplinary tool for the design of BWB aircraft and to demonstrate the performance of the tool on several example trade studies. The second objective is to develop a methodology for reducing the dimension of the design space using designer-chosen partitionings of the design variables describing the system.

The first half of this thesis describes the development of the BWB design tool and demonstrates its performance via a comparison to existing methods for the conceptual design of an existing BWB configuration. The BWB design tool is then demonstrated using two example design space trades with respect to planform geometry and cabin bay arrangement. Results show that the BWB design tool provides sufficient fidelity compared to existing BWB analyses, while accurately predicting trends in system performance.

The second half of this thesis develops a bi-level methodology for reducing the dimension of the design space for a trade space exploration problem. In this methodology, the designer partitions the design vector into an upper- and lower-level set, wherein the lower-level variables essentially serve as parameters, in which their values are chosen via an optimization with respect to some lower-level objective. This reduces the dimension of the design space, thereby allowing a more manageable space for designer interaction, while subsequently ensuring that the lower-level variables are set to "good" values relative to the lower-level objective. The bi-level method

is demonstrated on three test problems, each involving an exploration over BWB planform geometries. Results show that the method constructs surrogate models in which the sampled configurations have a reduction in the system objective by up to 4% relative to surrogates constructed using a standard exploration. Furthermore, the problems highlight the potential for the framework to reduce the dimension of the design space such that the full space can be visualized.

Thesis Supervisor: Karen E. Willcox
Title: Professor of Aeronautics and Astronautics

Acknowledgments

The research presented throughout this thesis was conducted with the support of many people. I would first like to thank my advisor Professor Karen Willcox for her high expectations and confidence in my work. She allowed a great deal of autonomy throughout this project while still pushing me towards my goals. Her sarcastic sense of humor provided great relief when I was most stressed during our meetings. I would also like to thank Dr. Andrew March for his continual encouragement and patience with any (stupid) question I might ask. He was always quick to respond to my often frantic emails and his seemingly infinite knowledge in both aircraft design and optimization was a vital resource for my success as a graduate student. I am forever grateful for all of your help.

I would also like to acknowledge the members of the ACDL community. The hilarious conversations and regular coffee breaks provided respite from the long hours spent in front of a computer screen. Between the football tossing with Patrick and Sergio's constant cynicism, my graduate student experience would not have been the same without this group (even if I was shoved into a corner of the worst building on campus for my first year).

I would like to thank all of the engineers at Boeing who helped make my work relevant and meaningful. Norm, Blaine, Zach, and Tony went out of their way to answer any technical questions I had concerning BWB design practices. They were accommodating and helpful during my visits to the Huntington Beach offices and really went out of their way in helping me fill in the finer details of my work. I am indebted to Dr. Liebeck for his belief in my work and influence in making it a reality. His visits to MIT helped guide me towards my ultimate goals while providing enough comic relief to last until the next visit. If my work provides the BWB group with only a small percentage of what I gained during this project, then my efforts have been worth it.

Most importantly, I would like to thank my friends and family for their unwavering support throughout both my undergraduate and graduate experiences at MIT. Six

years ago I moved away from Kentucky for the first time as a wide-eyed kid living in a big city for the first time. Today I have a slightly less pronounced southern accent but a much deeper appreciation for home. Any stress I was feeling would dissolve with a phone call home; the anticipation of a visit to Kentucky pulled me through several semesters. Throughout my years at MIT, a constant source of love and support was my beautiful girlfriend, Emily. The beginning of my undergraduate studies placed us some 1000 miles away, but that distance only seemed to strengthen our relationship. Though we've had some ups and downs along the way, I am ecstatic to call you my fiancée and I couldn't be more excited for our future. Thank you for putting up with my antics and always believing in me. I love you.

Finally, I want to dedicate this thesis to my Nanny Robb and Papaw Parks, who we may have lost over the past year but whose steadfast love remains with us today.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter describes the motivations for the work included in this thesis, as well as an overview of general layout of the thesis. Furthermore, a review of the literature which provided much of the motivation for the work is included. The chapter concludes with a statement of the objectives of this thesis.

1.1 Thesis Overview and Motivations

Multidisciplinary design optimization (MDO) is the use of numerical optimization techniques for the design of systems which contain a number of subsystems or disciplines. The primary motivation for using MDO is that system performance of a multisciplinary system is driven by the performance of the subsystems as well as the interactions among subsystems. Therefore, by utilizing MDO techniques early in the design process, designers can simultaneously improve their design and reduce the design cycle time. However, despite the advantages derived from employing MDO in system design, the general consensus is that "*...the genuine use of MDO methods within industry at-large is still rather limited*" [3].

The work performed in this thesis focuses on two issues in MDO which, if addressed, could bolster its appeal in industry. First, much of MDO's focus has been on algorithmic procedures for optimizing the decomposed system and not on the design process itself, thus there has been a neglect of human designers in the development of

MDO processes. Since engineering design, particularly at the conceptual design level, is an inherently human-driven process — through its reliance on human expertise and creativity in both the conception of the design problem and on the formulation of novel concepts to meet system requirements — a shift in focus towards a more human in-the-loop paradigm could bolster MDO's appeal to industry design. Secondly, the majority of distributed MDO architectures — that is, methods which decompose the overall system optimization problem into smaller optimization problems by decoupling the system along discipline lines — have not been demonstrated on actual systems or have known convergence issues on large, nonlinear design problems. The development of distributed methods is motivated by the structure of the engineering design environment, wherein discipline experts work on a system design nearly independently of the other groups comprising the system, and coordination of the full system occurs periodically. Thus, effective decomposition techniques which are representative of the real engineering design environment, or which could reduce large-dimensional design spaces to a more manageable size, would go a long way towards widespread adoption of MDO within industry.

This thesis is centered around the development of multidisciplinary methods to aid in the trade-space exploration phase of the design process for the blended wing body (BWB) aircraft. To address this objective, a multidisciplinary design tool for the conceptual design of BWB configurations is developed. The development of this tool places a particular emphasis on familiarity to existing BWB analysis tools and a reliance on the human configurator or discipline expert, both of which are achieved by providing an interface closely resembling existing disciplinary tools. Furthermore, the designer is explicitly placed in-the-loop of the design process in the BWB design tool, since it is developed as a graphical user interface (GUI) over which the designer has direct control. This tool simplifies geometry manipulation by defining a configuration via intuitive design variables and subsequently provides rapid configuration visualization as these design variables are manipulated. Low-fidelity discipline analyses are wrapped around this geometry engine which provide full-system simulations to generate quantities of interest. The multidisciplinary nature of the BWB design

tool allows experts from different disciplines to work independently in their disciplines of expertise while still incorporating those analyses from the other disciplines. Therefore, the BWB design tool provides a single tool from which any discipline can perform full-system trade studies.

Additionally, a framework is developed wherein the design space can be reduced by the designer such that trade studies may be performed more effectively. By partitioning the design variables into upper- and lower-level sets, the design space can be represented in terms of variables chosen by the designer. The lower-level variables are optimized with respect to some lower-level objective function — which is presumably easy to optimize relative to the system objective — while the remaining upper-level variables can be explored via trade studies. The reduction in the dimension of the design space allows the utilization of simplified visualization techniques. Furthermore, parallelizing this framework is straightforward, which can lead to a significant decrease in the time required to perform the trade study, as well as a better understanding of the design space.

1.2 Literature Review

The previous work underlying this thesis lies at the intersection between multidisciplinary approaches to decomposing the engineering system and techniques for driving the optimization of a system through guidance from the designer. Therefore, a discussion of several distributed MDO architectures is presented. Additionally, two approaches relying on visual steering techniques — in which the design space is reduced such that a designer can explicitly visualize and manipulate the design variables to gain insight into the design problem — are discussed. The last section discusses the Blended Wing Body aircraft and previous work which incorporated MDO methods into the design of the BWB.

1.2.1 Motivations for the Designer in-the-loop Paradigm

A major challenge in developing MDO methods is the tradeoff between reliance on the human designer and reliance on automated processes and algorithms to converge to an optimal design. While computers should certainly be utilized to bear the computational burden of exploring the often massive design spaces, the neglect of the human designer often has dire consequences on the ultimate design of the system. That is, automated codes often exploit gaps in the governing models of a design methodology and converge to impractical or even unrealizable designs. In fact, a 2010 NSF workshop on MDO amongst academic and industrial leaders in the field identified strategies to place designers "back in the loop" as a primary recommendation to advance MDO tools, even arguing that humans were better than computers in many aspects of the design and development process [38]. Thus, there exists a need to explicitly place the human into the MDO process.

There have been several approaches to placing the designer in-the-loop of the design process. The majority of these approaches rely on computational steering approaches drawn from the Scientific Visualization and Virtual Reality communities. That is, the computational steering paradigm allows a user to direct or "steer" a solution process to an answer faster by enabling the analyst to see the behavior of the variables during the analysis. The Visual Design Steering (VDS) paradigm developed by Winer and Bloebaum relies on graph morphing techniques to transform the design space such that a designer can visualize an n-dimensional optimization problem [47]. In this approach, the designer chooses to place two or three design variables on coordinate axes while the remaining design variables are placed on graphical switches. By adjusting the values of these switches, the designer can immediately view the impact of that design variable on the design space. This method uses a ranking procedure — whereby design variables are analyzed for their impact on the objective function and constraints via a sensitivity analysis — in an attempt to eliminate any potentially uninformative design variables from the design space [48].

Similar visual steering techniques were applied to trade space exploration, in which

a specialized tool, the Applied Research Laboratory Trade Space Visualizer (ATSV), was developed to allow users to sample interesting regions of the design space based on the calculation of attractors to user-defined preferences in the design space [43]. This tool allows a suite of visualization techniques for static data sets that have been generated offline, as well as interactive sampling and trade space exploration through direct query of the simulation model.

1.2.2 Decomposition Approaches for Multidisciplinary Systems

Engineering design for large-scale systems is generally not feasible for a single designer due to the requisite expertise in the many disciplines associated with the system [44]. Therefore, engineering design groups are comprised of discipline-specific experts, wherein the full system is independently designed by the different discipline groups and a consistent design is enforced via some system-level coordination strategy. To effectively model the design engineering environment and to capture the interaction effects between disciplines of a multidisciplinary system, it is often necessary to decompose the full system optimization problem along discipline lines into smaller optimization problems. This philosophy is the foundation for the development of distributed MDO architectures.

Practical approaches for decomposing large-scale systems were first studied in the 1960's as a means to partition mixed-variable programming problems via Bender's decomposition [6]. Soon thereafter, techniques for partitioning and "tearing" large linear systems via appropriate rearrangements and groupings of the system were investigated [42]. This work led to the development of the Design Structure Matrix (DSM), which has played an integral role in the visualization and development of decomposition approaches for MDO. The first MDO methods developed were monolithic architectures — where the MDO problem is solved by casting it as a single optimization problem — such as the All-at-Once (AAO) [9] and Simultaneous Analysis and Design (SAND) [17] architectures. While these methods are useful for solving

the multidisciplinary design problem, their inability to model the engineering design environment can limit their applicability to large-scale design within industry. However, because of their straightforward implementation, monolithic methods remain the popular approaches for the majority of MDO applications.

To enhance the appeal of MDO towards industry and to broaden the field's applicability, a consistent focus of the MDO literature has been on the development of methods to decompose the mathematical formulation of the design problem in order to mimic the structure of the engineering design environment; these architectures have been developed to replace the earlier-developed monolithic methods. One of the earliest distributed architecture is the Concurrent Subspace Optimization (CSSO) architecture, which decomposes the system problem into independent subproblems with disjoint sets of variables [39]. The first distributed architecture which fully decomposed the system to resemble the engineering design environment was the Collaborative Optimization (CO) method [8]. In this architecture, copies of the coupling and shared design variables are shared with all disicplines during each iteration of the solution procedure. Thus, each discipline subproblem is completely independent from the other discipline subproblems. Many other distributed methods have been developed which attempt to exploit the structure of engineering design problems. For example, the Bi-Level Integrated System Sythesis (BLISS) architecture assigns local design variables to discipline subproblems and shared design variables to the system subproblem [40]. A revised version of BLISS was developed, called BLISS-2000, which still relies on the formulation of surrogates for each subproblem but, similar to CO, uses coupling variable copies to enforce consistency at the optimum [41]. Therefore, the discipline subproblems can be run in parallel with minimal communication between disciplines.

Despite the prevalence of distributed architectures to solve MDO design problems, many of these methods have known convergence issues or, at the very least, converge significantly slower than the monolithic architectures that they were meant to supersede [28]. Therefore, the development of distributed approaches which provide robust convergence guarantees while allowing subsystem autonomy remains a

primary objective within the field of MDO.

1.2.3 The Blended Wing Body Aircraft

This thesis focuses on utilizing MDO techniques for the conceptual design of a Blended Wing Body (BWB) aircraft. Departing from the conventional tube-and-wing layout of traditional commercial aircraft, the BWB offered a significant increase in lifting area compared to a conventional aircraft, implying a substantial improvement in aerodynamic efficiency [22]. Consequently, the BWB has seen an influx of interest over the last decade due to its potential fuel burn savings and noise reduction capabilities [18].

Due to the strong synergy between the basic disciplines of the BWB, numerous attempts at integrating the design process through an MDO framework have been undertaken. Such efforts include the WingMOD code developed for Boeing [46] and the codes developed for the design of the SAX aircraft for MIT's Silent Aircraft Inititiative [11] [20]. Recently, the TASOPT code, while not strictly designed for industrial use, follows a similar paradigm of providing fully-automated optimization routines for the design of unconventional aircraft [14]. However, TASOPT's primary philosphy still rests in the automated optimization of the aircraft configuration, wherein a designer gives an initial configuration and the code independently performs the optimization. Thus, while these codes have significantly contributed to the understanding and design of BWB aircraft, their primary focus rested in the optimization of the configuration, instead of a means to effectively sweep the design space.

1.3 Thesis Objectives

Given the previous work related to multidisciplinary methods for trade-space exploration, along with the prior efforts to utilize MDO techniques for the design of the BWB, the objectives for this thesis are as follows:

1. Create a multidisciplinary analysis tool for the design of a blended wing body

aircraft which explicitly places the designer into the loop of the design process and can rapidly perform a full system simulation.

2. Demonstrate the performance of the BWB design tool through example trade studies of BWB configurations.

3. Develop a methodology for reducing the dimension of the design space by partitioning design variables of the system and demonstrate the method on an example multidisciplinary design problem using the BWB design tool.

1.4 Organization of the Thesis

The second chapter of this thesis will describe the methodology used in developing the discipline analyses for the BWB conceptual design tool, as well as the general layout of the tool. Specifically, this tool is built upon a geometry engine which serves as a user interface and allows rapid updates to the BWB configuration geometry; furthermore, the design tool integrates low-fidelity discipline analyses necessary to run a full system simulation. Chapter 3 demonstrates the performance of the BWB design tool on a baseline configuration via a walk through of a typical trade-study. The results of this trade-study are compared to a similar study performed for the Silent Aircraft Initiative at MIT. Chapter 4 of introduces a framework for decomposing the design space of a conceptual design problem in order to perform more effective trade-space exploration. An example problem is included to highlight potential utilization of the framework. Chapter 5 applies this framework to a BWB design problem. Using the BWB design tool, it is shown that appropriate partionings of the system design variables can lead to effective explorations over very few high-level design variables. Finally, Chapter 6 summarizes the thesis and provides suggestions for future research efforts for both the BWB design tool and system decomposition approaches.

Chapter 2

A Multidisciplinary Design Tool for the Blended Wing Body Aircraft

This chapter concerns the development of a graphically-based, multidisciplinary tool to aid in the conceptual design of the blended wing body aircraft, which will be referred to as the BWB design tool throughout the remainder of the thesis. The BWB design tool has been developed so that engineers from any discipline within the BWB design group can perform rapid trade studies by varying high-level parameters and running discpline-specific and full system analyses. Thus, the BWB design tool is developed to estimate aircraft performance quantities of interest — including fuel burn, takeoff gross weight, and cruise lift to drag ratio — and to balance the aircraft configuration. Because the performance of a BWB is strongly influenced by the interacting effects of discipline-level performance, it is crucial that the system analysis be performed in an integrated fashion. Furthermore, because of the objectives of this work, the majority of the discipline analyses are performed at a relatively low level of fidelity. Many parts of the discipline analysis routines are adapted from the well-known methods of conventional aircraft design [36] [35] [33]; however, for some parts of the analysis, it is necessary to adapt existing methods that are BWB-specific [11].

The BWB design tool uses Microsoft Excel as a foundation for its user interface.

Microsoft Excel allows easily rendered graphics which can be updated in real-time based on inputs to the configuration. Furthermore, Excel has a built-in Visual Basic API, which is used to interface with the variety of analysis codes used for the tool. The tool is decomposed along disciplines - and the disciplines are sometimes decomposed into sub-disciplines - through the use of worksheets. Thus, a full design loop is accomplished by simply stepping through each worksheet. The following sections describe each of the respective subsystem modules present in the BWB tool, including a description of the methodology behind each analysis, as well as a discussion of the level of fidelity of the respective analysis modules.

2.1 Layout of the BWB Design Tool

The BWB design tool is organized to model the typical process flow for the design of a BWB. Figure 2-1 shows the structure of the discipline analyses. The arrows correspond to information flow, thus it is clear that there is a significant utilization of the designer in the layout of the tool. Furthermore, the diagram shows the process for both calculating the two primary objectives of the tool: performance metrics of interest and balancing the aircraft.

The primary human interface with the BWB tool consists of several pages from which the designer can modify the geometry of the BWB. The configurator initially chooses a payload bay geometry via the 'CabinLayout' worksheet and wraps the chosen payload bay with an aircraft centerbody via the 'CenterbodyLayout' worksheet. Next, the designer sets the planform geometry outside of the centerbody by adjusting spanwise section parameters such as width, sweep, and chord via the 'PlanformLayout' worksheet. The 'ControlSurfaces' allows the designer to insert or modify control surfaces by varying inboard and outboard span- and chord-fraction, respectively. Finally, the 'ConstructionFeatures' worksheet provides the designer the ability to position the engines and landing gear, as well as to choose the locations of the outerbody spars (and thus the size and location of the fuel tanks). All designer adjustments are handled by hidden geometry worksheets, which have been developed to provide a

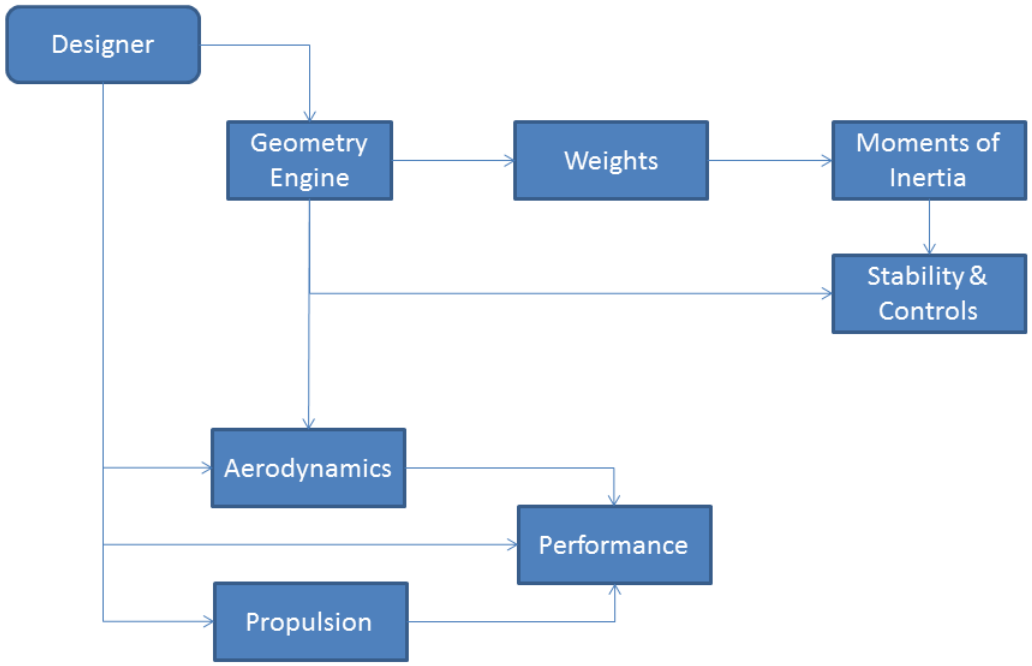


Figure 2-1: Process flow of the discipline analyses in the BWB design tool.

robust means to update the configuration geometry in real-time.

2.1.1 Geometry Parameterization

The BWB is an unconventional aircraft configuration and thus requires geometric definitions which may deviate from the usual aircraft vocabulary. Figure 2-2 shows the general geometry breakdown of a BWB configuration. The centerbody refers to the central portion of the aircraft which wraps the payload bays, while the outerwing refers to the traditional outerwing, as well as the blended region between centerbody and wing; in the BWB design tool, the blended region is decomposed into four 'kink' regions.

Central to the BWB design tool is the ability to easily update the geometry of a configuration, thus giving the designer rapid responses to geometry changes. The

Worksheet Name	Relevant Discipline	Function
'CabinLayout'	Geometry Engine	Define the shape of the payload bays
'CenterbodyLayout'	Geometry Engine	Wrap payload bays, alter centerbody geometry
'PlanformLayout'	Geometry Engine	Define OML
'ControlSurfaces'	Geometry Engine	Define elevon size and location
'ConstructionFeatures'	Geometry Engine	Define spar, engine, landing gear location
'TotalGeometry'	Geometry Engine	View full 3-D geometry
'Aerodynamics'	Aerodyamics	Run aero codes, view drag breakdown
'tetgen'	Aerodynamics	Generate 3-D mesh, run SU^2 Euler code
'Polars'	Aerodynamics/S&C	View/modify aerodynamic polars
'Weights'	Weights	Calculate weight of aircraft by functional groups
'MomentsOfInertia'	Moments of Inertia	Calculate MOI of aircraft
'Propulsion2'	Propulsion	Define engine properties
'S&CCases'	Stability & Controls	Calculate CG bounds of configuration
'Performance'	Performance	Simulates flight profile Calculates MTOW

Table 2.1: Worksheets present in the BWB design tool

geometry of a configuration is controlled by high-level design variables which affect spanwise cross sections of the planform; to describe the full three-dimensional configuration geometry, interpolation is used between the spanwise sections. The design-variables defining the geometry of each section are given in Table 2.2. As with all worksheets in the BWB design tool, yellow fields indicate design variables requiring designer input, while blue fields correspond to values which are automatically calculated by the tool. The majority of the design variables in the tool are located within the five worksheets corresponding to the geometry.

The BWB design tool describes the geometry as a finite distribution of spanwise cross sections. A two-dimensional planform based on this parameterization is shown in Figure 2-3. In particular, a total of 10 spanwise section — 5 sections which wrap the payload bay, thus describing the geometry of the centerbody of the BWB configuration, and 5 sections which describe the outerwing (four of which are called 'Kink' regions and the fifth is called 'Wing') — are used to describe the planform geometry of a BWB configuration. Moreover, at each section, the designer specifies a two-dimensional airfoil shape, thereby creating an airfoil stack which, upon interpo-

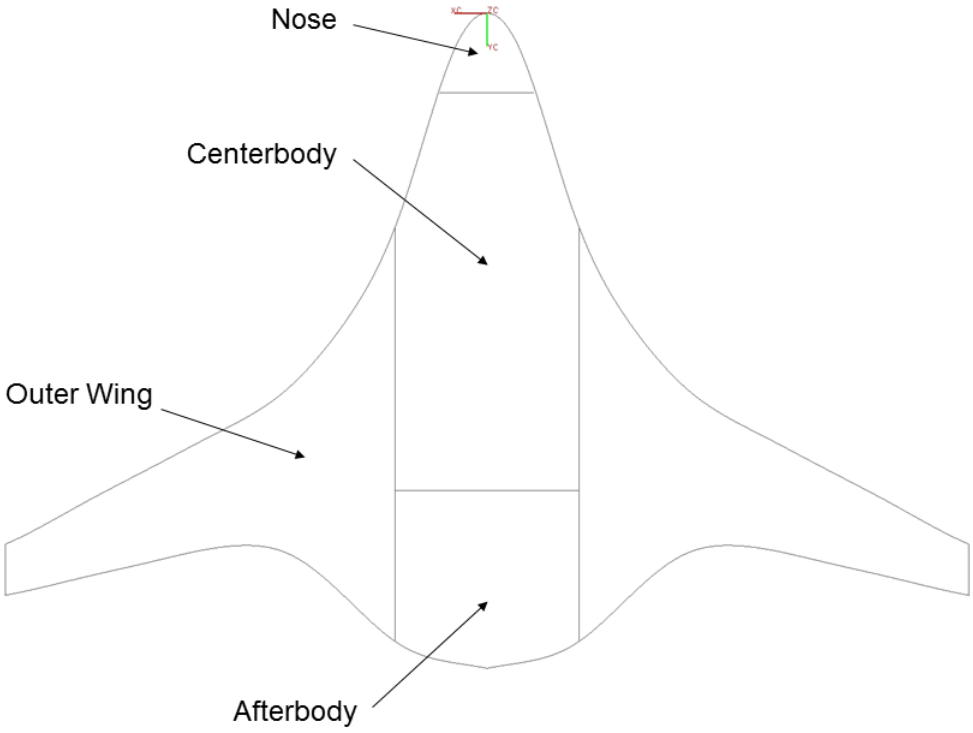


Figure 2-2: Geometric breakdown of BWB configuration.

Aircraft section	Variable name	Variable description
Centerbody	x_{LE}	x-location of the section leading edge
	z_{LE}	z-location of the section leading edge
	c	section chord length
	θ	section twist
	t/c	thickness-to-chord ratio
Outerwing	b	section width
	Λ_{LE}	leading edge sweep
	c_f	chord fraction
	c	chord length
	Γ	dihedral angle
	θ	twist angle

Table 2.2: Design variables describing each of the spanwise airfoil sections for the respective centerbody and outerwing of a configuration.

lation, describes the full three-dimensional aircraft geometry. Figure 2-4 shows this transformation from 2D airfoil stack to 3D geometry.

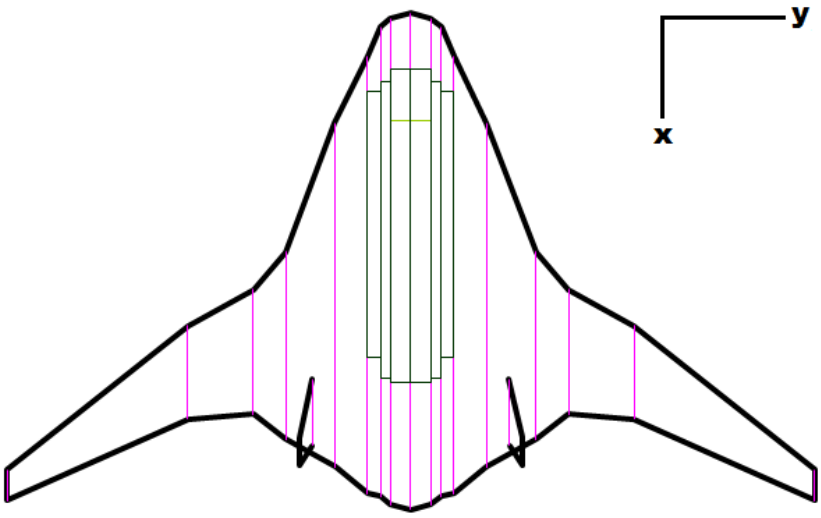


Figure 2-3: Spanwise section parameterization in BWB design tool shown by the vertical pink lines.

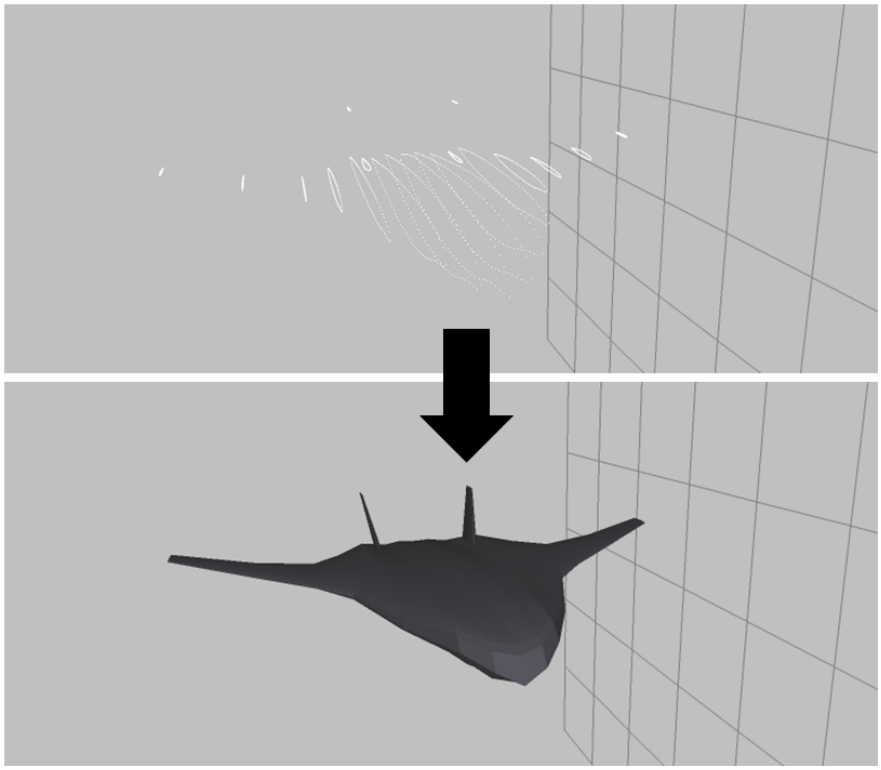


Figure 2-4: Transformation of the 2-dimensional airfoil stack to the full 3-dimensional geometry.

2.1.2 Airfoil Parameterization

The BWB design tool parameterizes the three-dimensional configuration as a spanwise distribution of airfoils, or an airfoil stack, which are specified by the human designer. For each spanwise cross-section, the designer specifies a two-dimensional airfoil shape via, at most, 41 normalized (x, z) points on the upper- and lower-surfaces, respectively. The current implementation assumes that the airfoils are generated outside of the tool, perhaps from a historical airfoil set or from a program such as XFOIL [12].

2.1.3 Geometry Worksheets Utilization

The primary user interface of the BWB design tool is the five worksheets which specify a configuration's geometry. These five worksheets are connected to a backend geometry engine which rapidly updates the configuration visualization mechanisms and are integrated with the various discipline analysis codes. All values in the geometry worksheets are given in inches and degrees; however, the plotting features allow any length scale, given it is used consistently. The following sections describe each of the worksheets - in chronological order - essential for developing a full BWB geometry using the BWB design tool.

'DefineCabins' Worksheet

To begin a new design of a BWB aircraft, it is necessary to first create a payload bay around which to design the aircraft. This is accomplished via the 'DefineCabins' worksheet. The shape of each cabin is described by values to the yellow input fields on this worksheet. For each cabin, the designer specifies the width of the cabins via the *c1width* variable, the sweep of the front and back walls (*c1LEsweep, c1TEsweep*), inboard length (*c1lengthIB*), height, and chamfer dimensions. The chamfer dimensions are designated by the width of the lateral chamfer on the inboard/outboard side of the cabin (*latchamberWIB/latchamberWOB*). The forward and aft chamfers refer to chord-wise distances. The depth of the chamfer (*ChamferD*) is assumed equal on each side.

Additional inputs vary between cabins. For the inboard cabin on the main deck, *XrefIB* sets the chord-wise coordinate of the front wall at the centerline. For each of the other cabins, *c2XoffsetIB* sets the chord-wise separation between the inboard edge of the front wall of the current cabin and that of the cabin inboard/above it. To set the vertical separation between the floor of a main deck cabins and the ceiling of the corresponding lower deck cabin, the *c1vertoffset, c2vertoffset,...* variables are used.

A designer can define up to four cabins on each side of the aircraft centerline, which are subsequently reflected across the x-axis. The floors of the all main deck cabins are assumed to be at $z = 0$. The side walls of the lower cabins are automatically aligned with those of the main deck cabins above. To create an odd number of cabins, the inboard chamfer width of the main deck cabin 1 should be set to zero and its width should be set to one half that of the desired center cabin. To remove both the main deck and lower deck cabins for a given pair, all inputs should be set to zero. To remove a cabin from only one deck, the width field value should be kept, while the lengths, leading and trailing edge sweeps should be set equal, and all other fields should be set to zero.

'CenterbodyLayout' Worksheet

Upon diagramming the cabin bays, the designer must fit airfoils around the centerbody at the location of each of the cabin's side walls. To use the "CenterbodyLayout' worksheet, the designer must first input (x,z) coordinates of the normalized airfoil for each cross section, as previously described. The blue columns to the right of each airfoil input field give the true (x,z) coordinates of the final airfoil, accounting for its chord, twist, thickness, and translation. These values can be modified by typing or scrolling values into the green input fields below the airfoil stack. Each airfoil can be modified by its leading edge position (*XLE, ZLE*), chord, twist, and thickness-to-chord ratio. Setting t/c to zero reduces the airfoil to its mean camber line. Changing the twist of an airfoil section rotates the airfoil about its leading edge.

The diagram above the adjustable inputs shows the cabins in cross section as well

as the airfoil being fit around them. A planform view of the aircraft is given to the right of the airfoil cross section plot. This allows the designer to fully wrap the cabin bays while simultaneously shaping the planform of the aircraft. To view the full 3-D geometry and check that the cabins are adequately wrapped, the 'TotalGeometry' worksheet should be accessed.

'PlanformLayout' Worksheet

The 'PlanformLayout' worksheet allows the designer to define the shape of the outer-wing section of the aircraft, as well as the inboard vertical tails. The right half of the spreadsheet contains input columns for the designer to enter airfoil coordinates for each of the edges of the outerwing sections, along with the inboard vertical tail. The shape of the aircraft outboard of the centerbody is divided into four 'kink' regions, a wing, and a tip on each side. The 'kink' regions allow the designer to approximate a smooth transition, or blend, between the centerbody section and wings.

For each section, the designer specifies its width, sweep, the chord fraction at which the sweep is measured ($cfraction$), the outboard chord ($chordOB$), dihedral, and the outboard twist ($twistOB$). The inboard values for a given section are defined by the outboard edges of the adjacent inboard section, such that the leading and trailing edges are continuous along the span.

The inboard vertical tails are defined by the (x,y) coordinates of the inboard edge, the length between the edges, sweep, and the inboard and outboard twist. The tail is assumed to be near vertical (i.e. dihedral near 90°) so that dihedral adjusts the (y,z) coordinates of the leading edge of the outboard airfoil relative to the inboard airfoil and the sweep changes the (x,z) coordinates of the leading edge of the outboard airfoil compared to that of the inboard airfoil.

To the right of the planform view is a front view that shows the maximum and minimum heights of the airfoils. Spin buttons located under this plot allow the designer to adjust the thickness of each airfoil's cross-section.

'ControlSurfaces' Worksheet

This worksheet allows the designer to place trailing edge control surfaces - generally elevons - along each of the spanwise sections. These control surfaces are defined by their inboard and outboard fraction of the span of the respective section at which their lateral edges are located. Additionally, the designer can specify the chord fraction at which the leading edge of the control surface will begin. To eliminate a control surface, each entry of that surface should be set to 1.

The right-side of the worksheet shows the airfoil coordinates of the interpolated airfoils located at the inboard and outboard edges of each control surface. These interpolations account for the translation, twist, chord, and thickness by computing a weighted average of corresponding airfoil coordinates on opposite edges of the component, using the respective span fraction of the control surface edge.

'ConstructionFeatures' Worksheet

It is in the 'ConstructionFeatures' worksheet where the engines, landing gear, spars, and fuel tanks are define. The current implementation of the BWB design tool can accommodate up to four engines. Defining an engine requires the designer to enter the coordinates of the center of the intake (X_{ref}, Y_{ref}, Z_{ref}), the length of the engine, its tilt (+ for increased lift) and angle (+ for toe-out), as well as the radius of the engine at both intake and exhaust ($rIntake$, $rExhaust$). To specify an odd number of engines, the designer must set one engine to be centered on the centerline of the aircraft. Zeroing all entries for an engine will eliminate that engine.

The landing gear consists of a nose and main gear defined by their overall dimensions. Both nose and main gear require the designer to enger their X_{ref} and Z_{ref} reference coordinates for the location of the mount, the length of the strut, the total width of the set of wheels ($wGear$), and the radius of the tires ($rTire$). By default, the nose gear Y_{ref} is set to zero so that the gear lies on the centerline; this value must be specified for the main gear. Additionally, the length of the main gear must be specified, which corresponds to the center-to-center distance between the front and

rear-most tires of the gear.

The *dirRetract* fields specify the way in which the gear retracts. For the nose gear, the options are 0° or 180° (towards the rear or front, respectively). For the main gear, the options are −90°, 0°, 90°, or 180° (inboard, rear, outboard, or front). A planform and front-end plot which includes the cabin and control surface outlines allow the designer to visualize any conflicts between the path of the gear and any existing components. This retraction is controlled by the scroll bar between the two diagrams.

The locations of the front and rear spar are defined by specifying the chord fraction of the centerline and outboard edge of each component. Values may range between 0 and 1. The spars can be visualized in the planform plot as the yellow lines. Additionally, spar locations define the fuel tanks. That is, fuel tanks can be specified for each of the ten possible regions bounded by the edges of previously defined sections. By checking the box to the left of the region number, the designer can assign a fuel tank to be placed in the region, bordered by the front and rear spars and the lateral edges of the corresponding section.

2.2 Aerodynamic Analysis

The aerodynamics module of the BWB tool allows a full drag buildup of a given aircraft configuration, as well as an estimation of a configuration's $C_{L_{\max}}$. A drag buildup was integrated into the tool using a variety of existing aerodynamic codes. From the aircraft configuration, the BWB tool can generate input files to Athena Vortex Lattice (AVL) [15], a panel method code, Boeing Panel Aerodynamics (Panair) code [24], and an Euler code, the Stanford University Unstructured (SU^2) open-source CFD code [32]. The integration of these codes into the BWB tool, along with the semi-empirical methods from [29] allows a full drag buildup for the aircraft configuration. The BWB tool computes the full drag-buildup automatically by clicking the "Run Drag Buildup" button on the 'Aerodynamics' worksheet. Moreover, both AVL and Panair can be run independently of a full-system simulation if the designer chooses.

The following sections describe the methodology behind the aerodynamic analysis present in the BWB design tool.

2.2.1 High-Speed Aerodynamics

Cruise drag buildup relies on multiple AVL runs at trim, as well as several empirical methods, to construct a drag polar consisting of three drag components. The drag buildup used in the BWB tool is estimated by,

$$C_D = C_{D_i} + C_{D_p} + \Delta C_{D_c} \tag{2.1}$$

where C_{D_i} is the induced drag, C_{D_p} is the profile drag, and ΔC_{D_c} is drag rise due to compressibility effects. Note that interference drag is neglected from the drag buildup.

Induced Drag

The induced drag is calculated using Athena Vortex Lattice (AVL) code [15]. The aircraft geometry is written to an AVL input file via a Visual Basic function and automatically exported to AVL. AVL then computes the lift and induced drag in the Trefftz Plane. Vortex lattice methods assume potential flow and that lift is always linearly proportional to angle of attack; thus, the method is quite accurate for low angles of attack.

Profile Drag

Profile drag is comprised of the drag due to both skin friction and form drag due to boundary layer growth. Both phenomena are predicted using well-known empirical relationships. Skin friction drag is treated as an empirical skin friction coefficient multiplied by the ratio of component wetted area to reference area, while pressure drag is modeled as a form factor multiplying the skin friction drag [29]. Wetted areas for each of the components are estimated using Panair. The formula for estimating profile drag is,

$$C_{D_p} = C_F \frac{S_w}{S_{ref}} FF. \tag{2.2}$$

The skin friction drag coefficient, C_F, is a weighted sum of the laminar skin friction and turbulent skin friction coefficients based on the fraction of the reference length exposed to each type of flow. The skin friction is computed by

$$C_F = C_{F_{\text{turbulent}}}|_l - \left(\frac{x_{\text{transition}}}{l}\right)\left[C_{F_{\text{turbulent}}}|_{x_{\text{transition}}} - C_{F_{\text{laminar}}}|_{x_{\text{transition}}}\right] \tag{2.3}$$

For commercial aircraft, $x_{\text{transition}}$ can be approximated quite accurately to zero [25]. The laminar skin friction coefficient assumes flat plate Blasius flow corrected for compressibility. Whereas for Blasius flow, the skin friction coefficient is only a function of the Reynolds number, this model uses a compressibility correction, C^*, the Chapman-Rubesin constant. Thus, C_F for the laminar case is given by

$$C_F = 2\left(\frac{0.664\sqrt{C^*}}{\sqrt{Re}}\right) \tag{2.4}$$

where the Reynolds number used for this method is the standard definition, given by

$$Re = \frac{\rho_{\text{ref}} V_{\text{ref}} L_{\text{ref}}}{\mu\text{ref}}. \tag{2.5}$$

Calculation of the turbulent skin friction coefficient is considerably more complicated, and relies on solving an implicit nonlinear system of equations involving the Reynolds number and compressibility effects. An in-depth discussion of the method is given in [29].

Form drag is closely related to the boundary layer momentum thickness, since it is the force caused by the boundary layer wake behind the body. The boundary layer momentum thickness is correlated to the skin friction coefficient through the integral boundary layer equations. Form drag is therefore a function of the body's thicknesses. Accordingly, the form factor is an empirical function of thickness to chord ratios for

planar aircraft components and fineness ratio for bodies of revolution. For wing-like shapes, the form factor is calculated by

$$FF = 1.0 + 2.7\left(\frac{t}{c}\right) + 100\left(\frac{t}{c}\right)^4 \tag{2.6}$$

where t/c is the thickness ratio of a particular component. An alternative form factor is used for bodies, given by

$$FF = 1.0 + 1.5\left(\frac{d}{l}\right)^{1.5} + 50\left(\frac{d}{l}\right)^3 \tag{2.7}$$

where d/l is the ratio of diameter to length (i.e. the reciprical of the fineness ratio).

Compressibility/Wave Drag

Drag increase due to compressibility effects appears at airspeeds higher than the critical Mach number, M_{cr}, due to shock waves. At Mach numbers slightly above M_{cr}, the drag increase is moderate, while higher Mach numbers (specifically, those above the drag divergence Mach number, M_{DD}) the drag slope experiences a sudden increase. To estimate the drag rise due to compressibility, or the wave drag, a 2-D method developed in [16] is used. This method calculates the drag-divergence Mach number for each of the 2 dimensional airfoil sections and applies a drag rise using empirical estimates. Because the airflow over a BWB is extensively 3 dimensional [18], this method only provides a rough estimate of the drag rise due to compressibility. A quasi-3D method employing MSES - similar to the analysis used for the design of the SAX-40 aircraft [18] was tested, but proved too computationally expensive for its limited accuracy. For accurate estimates of the compressibility drag rise on a BWB, a full 3-dimensional CFD code is required.

The method employed for the BWB design tool estimates the drag-divergence Mach number, M_{DD}, of each airfoil section as being the Mach number at which the drag-increase slope is 0.1. That is,

$$\left.\frac{\partial C_{dw}}{\partial M}\right|_{M=M_{DD}} = 0.1. \tag{2.8}$$

The drag rise can then be modeled using Lock's fourth power law [19]:

$$C_{dw} = \begin{cases} 0 & M \leq M_{cr} \\ 20(M - M_{cr})^4 & M > M_{cr} \end{cases} . \tag{2.9}$$

The contribution of the wave drag on a wing strip relative to the total three-dimensional wave drag is then calculated using the area ratio:

$$C_{Dw} = C_{dw} \frac{S_c}{S_{\text{ref}}}, \tag{2.10}$$

where S_c is the wetted area of the wing strip calculated by Panair.

The critical Mach number is estimated by

$$M_{cr} = M_{DD} - \sqrt[3]{\frac{0.1}{80}}. \tag{2.11}$$

where the drag-divergence Mach number is estimated for each wing cross section using the Korn equation extended with simple sweep theory. That is,

$$M_{dd} = \frac{\kappa_A}{\cos \Lambda} - \frac{(t/c)}{\cos^2 \Lambda} - \frac{c_l}{10 \cos^3 \Lambda}. \tag{2.12}$$

where κ_A is the airfoil technology factor, which is set to 0.95 at the suggestion of [16], and Λ is the mid-chord sweep of the section component.

The wave-drag calculation procedure thus divides the outerwing of a configuration into the pre-specified strips, as in the profile drag calculation. Each strip is therefore represented by its 2-D thickness ratio, lift coefficient, and half-chord sweep. The drag-divergence Mach number is then estimated via the Korn equation. Because this method requires an *a priori* estimate of the local lift coefficient, the spanwise lift distribution is taken from the Panair simulation to calculate wetted areas. Next, the local critical Mach number is estimated, and the subsequent cross-sectional wave-drag coefficient is found. Finally, the full 3-D wave drag is calculated by summing each of the outerwing sections.

2.2.2 Low-Speed Aerodynamics

To build drag polars for takeoff and landing, methods similar to those described above are used. The low-speed drag buildup is comprised of four drag components and is estimated by,

$$C_D = C_{D_i} + C_{D_p} + \Delta C_{D_{\text{gear}}} + \Delta C_{D_{\text{slats}}} \tag{2.13}$$

where C_{D_i} is the induced drag, C_{D_p} is the profile drag, $\Delta C_{D_{\text{gear}}}$ is the drag rise due to landing gear extension, and $\Delta C_{D_{\text{slats}}}$ is the drag rise due to slat extension.

Both the induced drag and profile drag are estimated using the same methods for the cruise analysis. The drag rise due to landing gear extension is empirically estimated to be twice the profile drag, although this value can be overridden by the designer. The drag rise due to slat extension is specified by the designer on the 'Aerodynamics' worksheet; the default value is taken to be $\Delta C_{D_{\text{slats}}} = 0.006$, as discussed in [35].

2.2.3 Estimation of $C_{L_{\text{max}}}$

The Valarezo method for estimating $C_{L_{\text{max}}}$ [45] is integrated into the BWB tool using a series of Panair simulations [24]. For a given configuration, the Valarezo pressure differential curve is scaled based on the Reynolds number at each spanwise section. Panair is then run on the clean configuration at several angles of attack to generate the maximum pressure differential curve at each spanwise section. The minimum differences between the scaled Valarezo curve and each pressure differential curve are then used to perform a line-fit to find the angle of attack at which the two curves first intersect. Panair is then run at the angle of attack found in the line fit to estimate $C_{L_{\text{max}}}$ of the clean configuration at low speed. Finally, this $C_{L_{\text{max}}}$ is scaled using the curve found in [35] to find the $C_{L_{\text{max}}}$ at cruise. Figure 2-5 shows an example plot generated by the low-speed aerodynamics module. The blue curve shows the scaled Valarezo pressure differential curve, while the other three curves represent the spanwise pressure differentials computed by Panair. $C_{L_{\text{max}}}$ is predicted to occur at the angle of attack of the pink curve, around spanwise section 10. The $C_{L_{\text{max}}}$ estimation

is extended to takeoff and landing using the $\Delta C_{L_{\max}} = 0.7$ factor given by Shevell for slat extension [36].

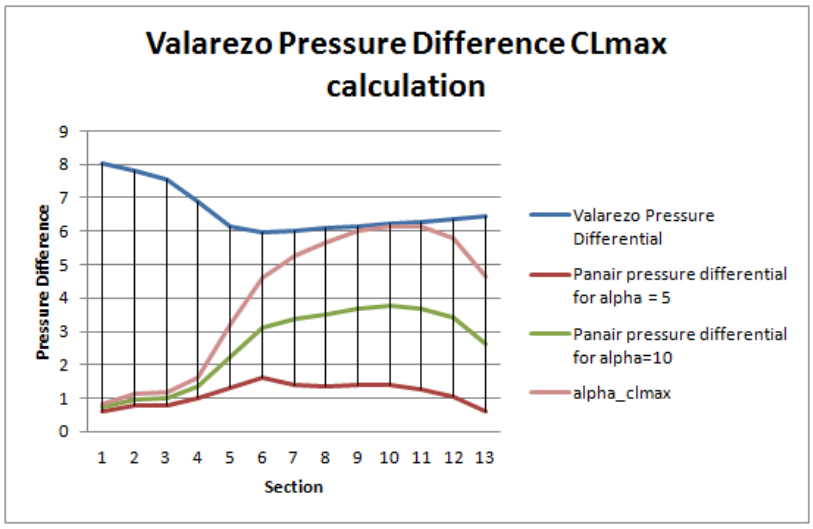


Figure 2-5: $C_{L_{\max}}$ estimation using the Valerezo pressure differential method.

2.2.4 Addition of SU^2 Euler Code

To increase the fidelity of the high-speed aerodynamics module, the Stanford University Unstructured (SU^2) open-source CFD code [32] is integrated into the BWB design tool. Specifically, a geometry mesh can be generated directly from the tool and an Euler code can be run on the configuration. Due to the geometry definition in the BWB design tool, the mesh generated from the tool is isotropic, thus having identical grid cells. Both the mesh generation and the Euler code can be run from the 'tetgen' worksheet within the BWB design tool. Due to the higher fidelity of this analysis and the subsequent increase in computational time, this code is left out of the simulation loop. However, the analysis is still included in the tool and can be run on a case-by-case basis via the 'Run CFD' button on the 'tetgen' worksheet.

2.3 Weights and Moments of Inertia Analysis

An analysis module to estimate the operating empty weight (OEW) of an aircraft configuration was provided by Boeing. The analysis uses empirical weight curves

based on configuration geometry to estimate the quantities of interest. This module is automatically updated when the geometry is changed and thus requires no user input. Furthermore, the module provides a weight breakdown into approximately 20 functional groups. The functional weight groups are shown in Table 2.3. Additionally, the performance module iterates on the maximum takeoff weight (MTOW) of the aircraft to solve a designer-specified sizing mission.

Functional Group	Mass Breakdown for MOI
Outerwing	Distributed across outerwing
Centerbody	Distributed across centerbody
Afterbody	Distributed across afterbody
Winglet/Vertical Tails	Distributed across vertical tails
Landing Gear	Point mass
Engine System	Point mass at pylon location
Fuel System	Distributed across fuel box (defined by spar location)
Flight Controls & Hydraulics	Distributed across nose
Electrical	Distributed across cabins
Pneumatics & APU	Point Mass
Anti-Icing	Distributed across wings
Furnishings & Equipment	Distributed across cabins
Instruments	Distributed across nose
Avionics	Distributed across nose

Table 2.3: Functional weight groups for the weights module.

The weights module was extended to estimate the center of gravity (CG) and moments of inertia (MOI) of a configuration. To estimate the CG of a configuration, the CG of each functional weight group is estimated and translated to its location on the aircraft, and the CG is found by

$$\mathrm{CG} = \sum_{i=1}^{N} \frac{\mathrm{CG}_i W_i}{W}. \tag{2.14}$$

The MOI analysis integrated into the BWB tool is adapted from the classic method developed by Marsh [27]. The method breaks the full aircraft configuration into trapezoidal sections of linearly increasing thickness – an inherent characteristic of the geometric parameterization of the BWB tool – and proportions the functional weight groups to each section, either in a distributed sense or as a point mass, depending

on the nature of the functional group. The centroid and MOI for each trapezoidal section are then computed using the equations in [27]. Finally, each section MOI is translated to the CG of the aircraft to estimate the full configuration MOI.

2.4 Stability and Controls Analysis

To estimate the forward and aft CG-limits, and thus to determine if the configuration is balanced, a stability-and-controls (S&C) module is integrated into the BWB design tool. The S&C analysis methodology is adapted from existing methods provided by Boeing and relies on AVL simulations, along with simply force balance equations. The module estimates the CG envelope based on the following longitudinal requirements:

1. Forward CG-limit requirements
    (a) Trimmed stall speed = Landing Speed/1.23 (nose-up long control power)
    (b) Nosewheel liftoff at 3.0 deg/s^2 pitch acceleration (nose-up long control power)
    (c) Go-around at 6.0 deg/s^2 pitch acceleration (nose-up long control power)
    (d) Nosewheel hold-off on ground at stall speed (nose-up long control power)
2. Aft CG-limit requirements
    (a) Stall recovery at -4.0 deg/s^2 pitch acceleration (nose-down long control power)
    (b) Nosewheel steering with $>$ 4.0% weight on nose gear (main landing gear placement)
3. Unaugmented stability requirements
    (a) $>$ 0% static margin at design cruise Mach number

The static stability requirement is checked using a single AVL run. Since the aircraft CG is calculated in the weights analysis module, it is necessary to estimate

the neutral point of the configuration at the design cruise Mach number. Thus, a single AVL simulation at the design condition gives the neutral point. If the CG is forward this neutral point, the configuration is statically stable.

Calculation of the forward and aft CG-limits given the above requirements is slightly more involved. The BWB is controlled via elevons which may span the trailing edge of the entire geometry, where the size and location of each control surface is set by the designer on the 'ControlSurfaceLayout' worksheet of the BWB design tool. The AVL simulations which are used to build the drag polars also calculate the requisite control derivatives for each surface. Once the drag polars are constructed, a system of equations is solved iteratively such that the moments on the aircraft sum to zero. The CG bounds are then chosen to be the maximum CG calculated for the forward limits and the minimum CG calculated for the aft limits. The aircraft is balanced if the CG calculated from the weights module is located within the bounds calculated for the S&C analysis.

2.5 Performance Analysis

A performance model is included in the BWB design tool which solves both a sizing mission and a reference mission. The specific performance model integrated into the BWB design tool was adapted from a preexisting, low-fidelity Boeing model, which uses methods similar to those described in [14]. Given a BWB configuration, the performance model builds appropriate drag polars and calculates the flight profile and fuel burn for the specified mission using a simple stepwise integration. Figure 2-6 shows the mission profile calculated by the performance module for a baseline configuration.

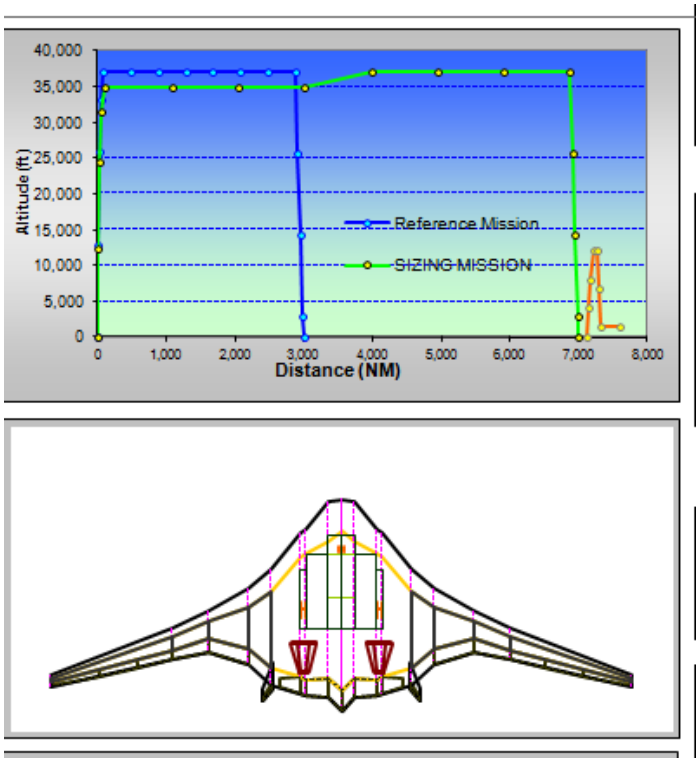


Figure 2-6: Flight profile for the shown BWB configuration calculated by the performance module of the BWB tool. The blue profile shows a short-range reference mission while the green profile shows a long-range sizing mission.

Chapter 3

Demonstration of BWB Design Tool Performance

To analyze the methodologies used for the discipline analyses in the BWB design tool, the performance of the tool on a BWB configuration resembling the SAX-40 aircraft designed for the Silent Aircraft Initiative (SAI) by researchers at MIT [18] is demonstrated. The analysis performed for the SAI on the SAX-40 is used as a benchmark on which to compare the results of the BWB design tool. The BWB design tool is used to conceptually design the SAX-40 aircraft configuration using models of nearly equal fidelity to those contained in the SAI analysis; therefore, the performance metrics calculated by the BWB design tool can be appropriately validated against a prior detailed analysis of the configuration. Furthermore, a comparison of the individual discipline analyses is shown. Comparing the two analyses, it is shown that the BWB design tool calculates the performance quantities of interest to within approximately 5%, although the calculated fuel burn has a slightly higher discrepancy due to different propulsion models. Further, the cruise aerodynamics analysis of the BWB design tool underpredicts lift and overpredicts drag relative to the SAI analysis.

In addition to a comparison of the results of the BWB design tool with the SAX-40 aircraft, two example trade studies are performed using the BWB design tool. These example trade studies highlight two of the potential uses of the BWB design tool, namely the tool's ability to rapidly update the configuration geometry and

analyze system performance. Results of the trade studies show that the BWB design tool predicts the correct trends in performance quantities of interest for changes in the planform geometry. Moreover, the trade studies provide direction into the best performing planform geometries and cabin bay arrangements.

3.1 Sample Aircraft Configuration

The SAX-40 geometry, as designed by the SAI study, is shown in Figure 3-1. This configuration uses an airfoil stack consisting of four airfoils which are distributed along the span of the aircraft as shown in the figure.

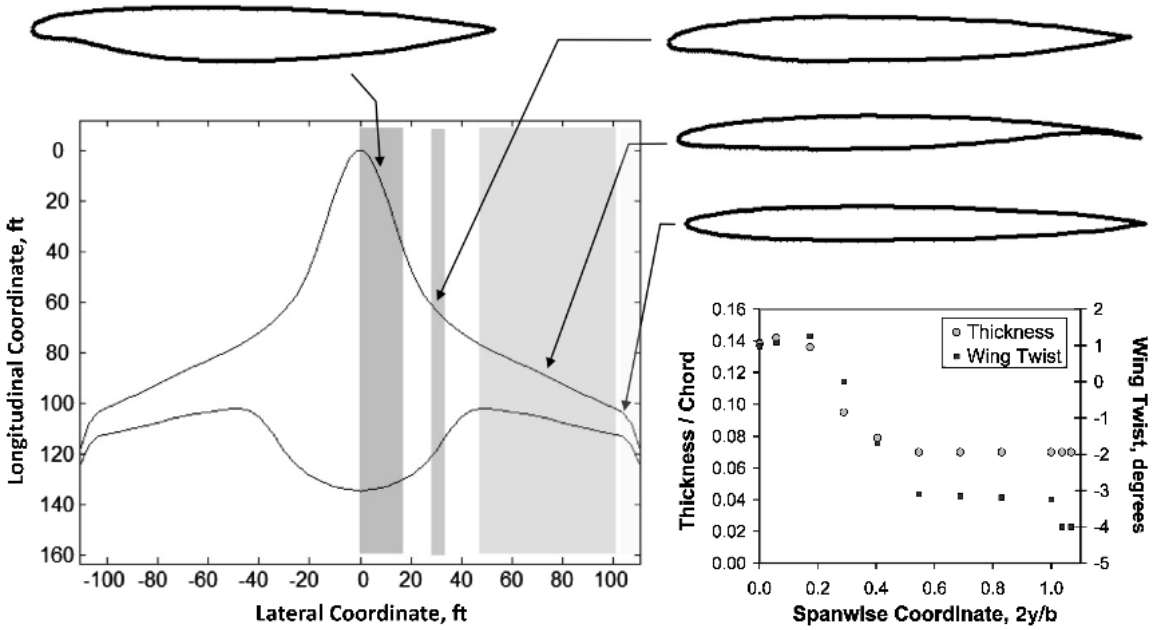


Figure 3-1: Geometry of the SAX-40 configuration (taken from [18]).

Differences in the geometric parameterization between the BWB design tool and the SAI tools do not allow a direct import of the SAX-40 geometry. Therefore, the original SAX-40 geometry is partitioned into 8 spanwise sections, which are used to describe the planform geometry in the BWB design tool. Figure 3-2 shows the resulting geometry given by the BWB design tool. The four airfoil sections which are used for the SAX-40 were obtained and appropriately apportioned to each spanwise section, as noted in the figure.

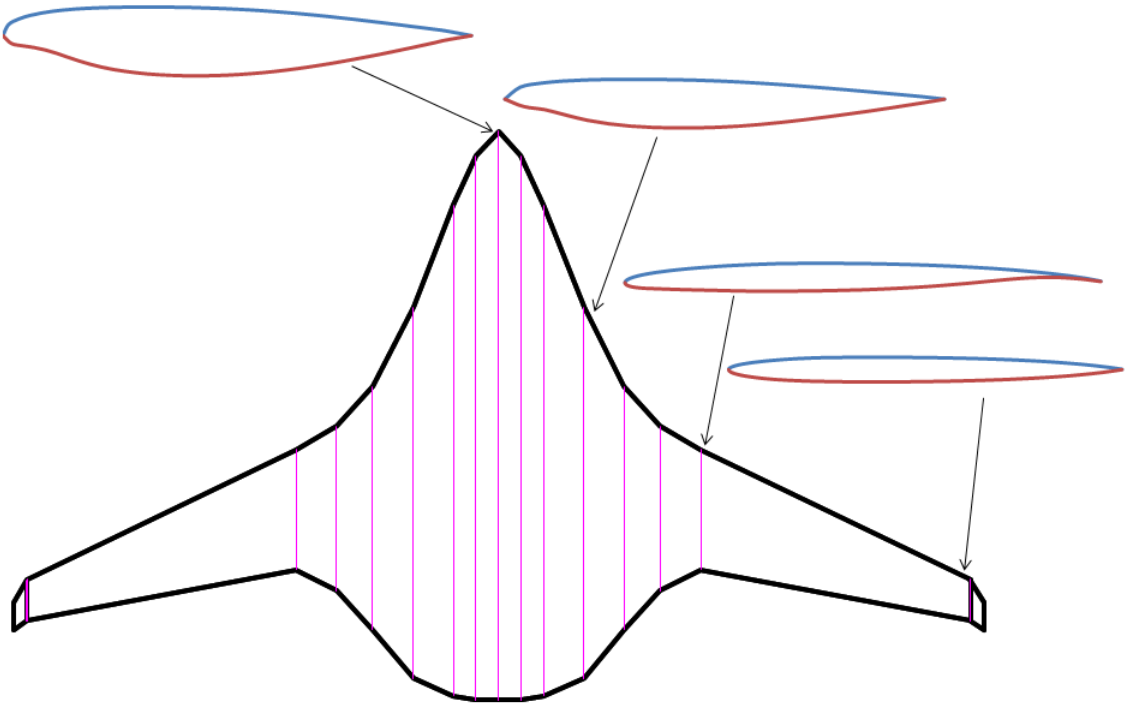


Figure 3-2: Geometry of the SAX-40 configuration given by the BWB design tool. The arrows indicate the first spanwise section which uses the indicated airfoil. The airfoil is used for each subsequent planform section until a new airfoil is specified.

The geometric parameters of the SAX-40 are given in Table 3.1, along with the quantities calculated by the BWB design tool; this table also shows the nominal mission for which the comparison was made. The table highlights the similarity in the geometric models used, as both the planform area and the wing span of both models are nearly identical. However, because the true SAX-40 geometry could not be explicitly input into the BWB design tool, there still exist slight geometric discrepancies, which can have quite significant effects on the drag computed for each configuration, as well as the computed performance metrics. This is discussed in the following section.

Parameter	SAI value	BWB design tool value
Planform area, ft^2	8998	8997
Wing span, ft	207.4	207.5
Cruise Mach	0.8	0.8
Range, nmi	5,000	5,000
Payload, lb.	51,600	51,600

Table 3.1: SAX-40 geometric parameters

3.2 Individual Discipline Comparison

Using the geometry of the SAX-40 configuration as described above, the aircraft performance predicted by the BWB design tool is compared to the performance values of the SAI analysis as shown in [18]. The following sections show a comparison between the relevant performance parameters, as well as a breakdown of aerodynamic and weight parameters, respectively. Additionally, a discussion of any discrepancies between the two analyses is provided. Results show that the BWB design tool overpredicts the cruise drag relative to the SAI analyses due to a larger estimate of induced drag. The OEW computed by the BWB design tool is slightly larger than that computed by the SAI, primarily due to model discrepancies stemming from assumptions on advanced technologies used in the SAI study. Similarly, there is a fairly large difference in estimated fuel burn, primarily caused by the SAI analyses using a lower SFC due to their use of a boundary layer ingestion model. Finally, the performance metrics of interest calculated by the BWB design tool are shown to be within an acceptable accuracy given the expected level of fidelity.

3.2.1 Aerodynamics Analysis Comparison

A comparison between the beginning of cruise drag buildup of the BWB design tool and SAX-40 analysis is shown in Table 3.2. The SAX-40 drag buildup relied on AVL and MSES, a 2-D compressible, viscous airfoil design and analysis tool [13], along with empirical relations. This drag estimation method was validated against simulations from CFL3D, a 3-D Navier-Stokes solution, and was found to overpredict the lift-to-drag ratio of a configuration, primarily due to the highly 3-D flow near the centerbody.

A thorough analysis of this validation procedure and the resulting discussion are found in [18]. Despite the potential discrepancy of the SAX-40 drag prediction, it was noted that the methodology adequately captures the 3-D aerodynamic features for the purposes of conceptual design; therefore, a comparison between these methods and those used for the BWB design tool are sufficient given the aims of the design tool.

A comparison between the spanwise lift distributions at an angle of attack of 5 degrees computed by AVL for the configuration used for the BWB design tool and a full 3-D Navier-Stokes solution of the SAX aircraft from the SAI analysis (found in [34]) was peformed. The lift curves are similar for both cases, exhibiting a nearly elliptic spanload and significant lift at the centerbody. Given the similarity of the profiles, the use of AVL to build the lift polars suffices for the BWB design tool.

Additionally, a drag breakdown at the begininning of cruise for the given mission was calculated by the BWB design tool; like the SAI analysis, the BWB design tool builds the total drag from three drag components, including profile drag, induced drag, and wave drag. Table 3.2 compares the drag breakdowns of the SAX-40 configuration between the BWB design tool and the SAI analysis. The overprediction in the cruise lift coefficient is most likely due to the larger MTOW calculated by the BWB design tool, which is explained in the following section. Moreover, the BWB design tool overpredicts the total drag on the SAX-40 by roughly 1 drag count, or about 1%. Both the profile drag and the drag rise due to compressibility are estimated almost identically to the values calculated for the SAI, whereas the induced drag is overpredicted by roughly 4%. The similarity in profile drag prediction is expected, given that similar wetted area methods were used for both the BWB design tool and the SAI analysis. Furthermore, the similarity in wave drag verifies that the methodology used for the BWB design tool, while quite different and of a lower fidelity than the SAI method, suffices for the aerodynamic analysis. The discrepancy in induced drag is a bit alarming considering both analyses used the same calculation procedure, namely AVL. However, since induced drag is primarily a function of the aircraft geometry and, in particular, quite sensitive to the twist and thickness distri-

butions, the discrepancy in induced drag is rooted in small geometric differences in the SAX-40 configuration used in the BWB design tool. A sensitivity analysis on the effect of twist on the induced drag of the baseline BWB configuration showed that a 0.1° decrease in outerwing sweep increases the induced drag computed by AVL by 3.3%, or nearly a full drag count. Therefore, given that the induced drag is quite sensitive to the outerwing twist, the aerodynamic analysis of the BWB design tool is sufficient given the objectives of the tool.

Coefficient	BWB design tool value	SAX-40 value	% difference
C_L	0.2073	0.2205	6.4
C_{D_i}	0.0025	0.0024	4.2
C_{D_p}	0.0054	0.0054	0.0
C_D wave	0.0001	0.0001	0.0
C_D	0.0080	0.0079	1.3

Table 3.2: Comparison of the aerodynamic coefficients at beginning of cruise between the BWB design tool and the SAX-40.

3.2.2 Weights Analysis Comparison

The full-system simulation from the BWB design tool provides a weight breakdown similar to the analysis for the SAX-40. Table 3.3 shows a comparison between the two weight breakdowns. The MTOW calculated by the BWB design tool is almost 5% larger than the MTOW calculated by the SAI. Furthermore, the OEW estimated by the BWB design tool is roughly 6% greater than that from the SAI analysis, showing that the empirical weight buildups for both analyses are similar, though discrepancies do exist. The deviation in MTOW stems from the overprediction of both the OEW and the fuel burn from the BWB design tool. The nearly 6% overprediction in fuel burn predicted by the BWB design tool is primarily due to a larger cruise SFC. Whereas the propulsion model of the BWB design tool calculated a cruise SFC of 0.54 lb/lb-hr, the SAI analysis included the effects of boundary layer ingestion and thus had a reduced cruise SFC of 0.49 lb/lb-hr. Therefore, the larger fuel burn calculated by the BWB design tool is due to the larger estimated OEW, along with the use of less efficient engines. However, despite the discrepancy in fuel burn calculation,

the following sections of this chapter confirm that the BWB design tool predicts the appropriate trends in planform geometry changes on fuel burn. Thus, the MTOW calculation procedure used in the BWB design tool suffices for the objectives of the tool.

Component	Mass from BWB design tool, lb.	Mass from SAX-40, lb.	% difference
Structure	101,823	104,870	2.9
Fixed equipment	57,978	51,220	13.2
Landing gear	23,562	14,760	59.6
Propulsion	36,423	36,810	1.1
OEW	219,786	207,660	5.8
Design payload	51,600	51,600	0.0
Fuel with reserves	77,606	73,310	5.9
MTOW	348,643	332,560	4.8
Weight fraction	0.65	0.62	4.8

Table 3.3: Comparison of the weight buildup modules for the BWB design tool and the SAX-40.

3.2.3 Performance Analysis Comparison

A full system performance analysis was run on the SAX-40 geometry using the BWB design tool. Table 3.4 shows a comparison between the performance metrics computed by the BWB design tool and those computed by the SAI analysis.

	SAX-40		BWB Design Tool	
Parameters	Begin cruise	End cruise	Begin cruise	End cruise
Cruise altitude, ft	40,000	45,000	40,000	42,000
Lift coefficient	0.2064	0.2091	0.2205	0.2013
C.g., % centerbody chord	58.3	57.1	58.2	57.4
Static margin, %/in.	5.9/31	9.5/50	8.5/44	11.0/57
ML/D	20.1	18.8	21.8	20.7

Table 3.4: Comparison of the aerodynamic performance parameters between the BWB design tool and the SAX-40.

Since both analyses used AVL to compute the neutral point of the aircraft in order to estimate the static margin, the discrepancy in CG helps explain the discrepancy in static margin. The differences in the estimated CG is primarily affected by these

weight discrepancies. Specifically, the overprediction of both landing gear weight and fixed equipment weight, along with the distribution of these weights throughout the aircraft, explains the discrepancy in both CG and static margin. That is, because the BWB design tool estimates a significantly greater weight for the landing gear than does the SAI analysis, the CG is more sensitive to the gear placement. Regardless of the absolute error in static margin prediction, the BWB design tool did predict the SAX-40 configuration to be statically stable, which suffices for the purposes of the tool.

Apart from the static margin estimation, the BWB design tool estimates the other quantities of interest to within roughly 10% relative to the SAI analyses. The general agreement in performance estimates between the BWB design tool and the SAI analyses shows that the level of fidelity included in the BWB design tool is sufficient for conceptual design, since both methods were ultimately developed for conceptual design. The discrepancies in the predictions primarily stem from the assumptions about advanced technologies used for the SAI analyses, which are not represented in the BWB design tool. Specifically, the SAI methods were developed with an emphasis on reducing aircraft noise, whereas the BWB design tool is focused on the exploration of current BWB designs for fuel burn and aerodynamic performance. Thus, the tools serve different objectives. Moreover, the absolute values calculated by the BWB design tool are not nearly as important as the accuracy of the trends estimated by the tool, which are analyzed in the following sections.

3.3 Example Trade Studies Using BWB Design Tool

The primary objective in the development of the BWB design tool is to provide a designer the ability to perform full system analysis of different BWB configurations. During conceptual design, a variety of drastically different planform geometries may be analyzed in an attempt to search a wide portion of the design space. Furthermore,

because the BWB is an unconventional aircraft with few baseline configurations to compare against, the design space is not well understood, thereby magnifying the importance of trade studies. The following section shows two example trade studies which aim to represent realistic trade studies during conceptual design. Both trade studies focus on design spaces which include aspects of the planform geometry, although utilization of the BWB design tool for performing trade studies is certainly not limited to only studying changes in configuration geometry. The first example trade study analyzes the effects of changes in high-level geometric variables describing the planform geometry on performance while the second example trade study analyzes the effects of the number of cabin bays on performance.

3.3.1 Planform Geometry vs. Performance

To assess the capabilities of the BWB design tool for performance trend estimation with respect to planform geometry changes, this example study analyzes several BWB planform geometries using the BWB design tool. This exercise also serves as a means to check that the analyses contained within the BWB design tool are appropriately predictive, since such trends are generally well understood. Thus, for a specified mission — in particular, for the cruise Mach number and payload fixed at the values used for the SAX-40, along with a range of 4,000 nautical miles — the planform geometry is modified by changing the outerwing section widths and sweeps while keeping the SAX-40 centerbody and fixing the total wingspan to 207.5 ft. Specifically, the design variables parameterizing the 3 outermost spanwise sections (referred to as 'Kink3,' 'Kink4,' and 'Wing' on the PlanformLayout worksheet) are changed in order to modify the planform geometry. The effects of these changes on OEW, fuel burn, and cruise ML/D are calculated. Additionally, for this problem, the airfoil sections are kept fixed to those used by the SAX-40, since this airfoil stack has been designed specifically for BWB configurations.

A total of six configurations are analyzed for the trade study. Figure 3-3 shows the planform geometries of each of the analyzed configurations. Furthermore, Table 3.5 gives the values of the outerwing section design variables for each of the analyzed con-

figurations. Changes in sweep and section widths account for the primary differences in planform geometry.

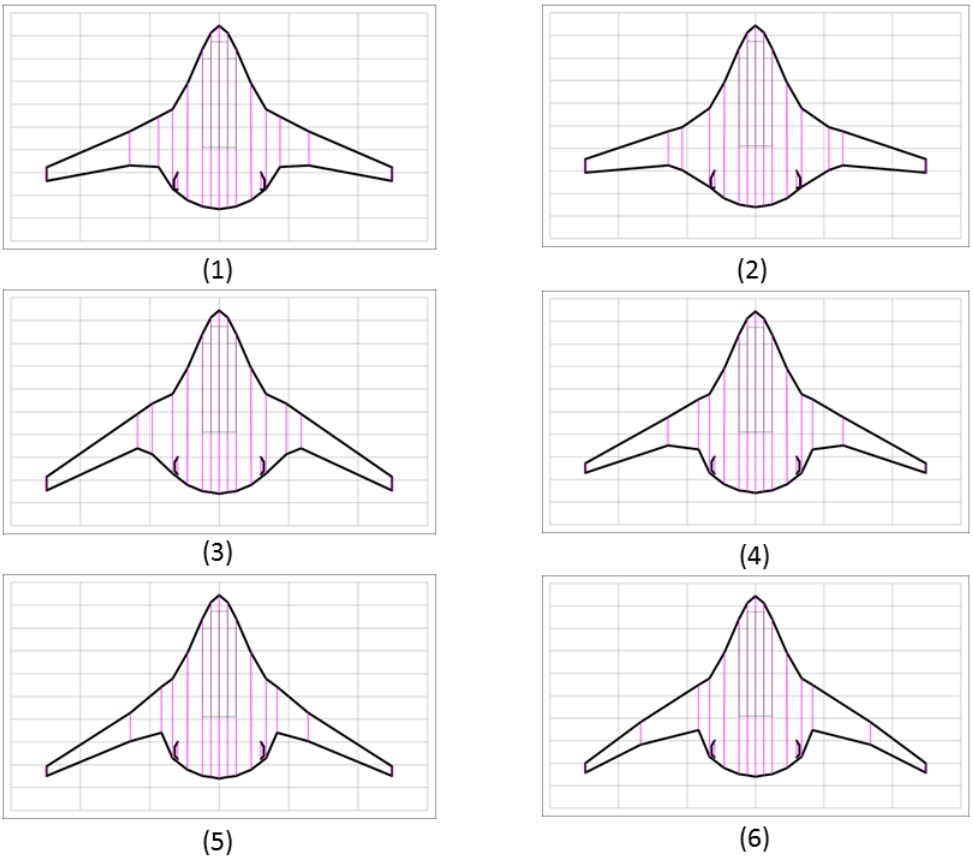


Figure 3-3: The six BWB configurations used for the trade study. The total wingspan is equal for all configurations.

Discussion of Results

Results of the full-system simulations are shown in Table 3.6. From the simulations, general trends predicted by the BWB design tool due to planform geometry changes can be extracted. Consider, for example, configurations 1 and 2. Configuration 2 has a slightly larger planform area due to a wider Kink3 section; this increased planform area causes an increase in the OEW of configuration 2, as expected. Moreover, configuration 2 has a considerably reduced wing sweep relative to configuration 1. Similar to conventional aircraft, this reduction in wing sweep results in a decrease in cruise ML/D, along with a resultant increase in fuel burn, primarily due to an

Config.	Outerwing Section	Width, in.	Chord, in.	Sweep, deg.	S_{ref}, ft.^{2}
1	Kink3	101.0	442.0	32.0	9246
	Kink4	208.0	300.0	32.0	
	Wing	600.0	120.0	27.8	
2	Kink3	201.0	380.0	40.0	9372
	Kink4	100.0	300.0	20.0	
	Wing	608.0	120.0	22.0	
3	Kink3	146.0	442.0	30.0	9246
	Kink4	108.0	300.0	40.0	
	Wing	655.0	120.0	40.0	
4	Kink3	80.0	442.0	30.0	8741
	Kink4	224.0	250.0	35.0	
	Wing	605.0	85.0	34.0	
5	Kink3	80.0	400.0	28.0	8653
	Kink4	224.0	250.0	45.0	
	Wing	605.0	85.0	38.0	
6	Kink3	80.0	400.0	20.0	8803
	Kink4	424.0	200.0	38.0	
	Wing	405.0	85.0	42.0	

Table 3.5: Geometric design variables of the modified outerwing sections for each BWB configuration.

increase in compressibility drag. Similarly, a comparison between configurations 5 and 6 shows the BWB design tool's predictions on the effects of outerwing sweep. These two configurations are almost identical, except for an increase in Kink4 section width and an increase the outermost section sweep. As expected, the configuration with greater wingsweep results in a higher cruise ML/D and a subsequent decrease in fuel burn.

Apart from outerwing sweep, effects of taper ratio and aspect ratio changes can be extracted from the trade study. Configuration 4 has a reduced taper ratio relative configuration 1. Configuration 4, although having a lower OEW and slightly greater outerwing sweep, has a lower cruise ML/D. This occurs because of the reduced lifting capability of the outerwing due to the tip chord being too small. Comparing configurations 3 and 4, effects on outerwing aspect ratio can be analyzed. Specifically, configuration 3 has a smaller aspect ratio relative to configuration 5. As expected, configuration 3 has a lower cruise ML/D primarily due to increased induced drag

Config.	Static margin (beg. cruise), %	OEW, lb.	Fuel burn, lb.	Cruise $M\frac{L}{D}$
1	2.7	210,771	64,773	19.95
2	3.5	211,587	65,803	19.76
3	5.2	211,003	68,059	19.20
4	2.0	206,029	73,034	19.44
5	10.8	205,694	64,046	19.77
6	6.1	207,473	63,065	20.13

Table 3.6: Performance quantities of interest calculated for each of the configurations.

due to the lower aspect ratio.

While this example study shows a comparison between a small number of planform geometries, the chosen configurations represent a fairly large sweep across the BWB design space for the specified mission. Moreover, the methodology for this example can be easily extended to many more geometries using the BWB design tool. Thus, while this trade study is somewhat simplified relative to a conceptual trade study performed for a real BWB design, this example study shows that the BWB design tool correctly predicts trends in planform geometry changes and, moreover, provides the ability to rapidly analyze a wide array of BWB configurations.

3.3.2 Number of bays vs. Performance

A trade study which is relevant to efforts to commercialize the BWB is the number and positioning of the cabin bays in the centerbody of a configuration. The cabin bays of a BWB are wrapped within the centerbody of the aircraft — which is generally wider than a conventional aircraft — and can therefore be distributed in a number of ways. Similar to a conventional aircraft, a single long cabin bay can be placed along the centerline of a BWB, causing the planform to closely resemble a conventional tube-and-wing design; this configuration resembles early efforts by Boeing to design BWBs [22]. This trade study examines BWB configurations involving one, two, three, and four cabin bay arrangements and the resulting planform geometries which arise. Effects of each of the cabin bay arrangements on the performance of the aircraft, including OEW, weight fraction, and fuel burn, are analyzed.

For this example study, an A321 baseline configuration is used to size the cabin

bays, along with its nominal mission. Table 3.7 shows the cabin dimensions and mission characteristics for a single class configuration. Given the baseline characteristics, cabin bays which are configured with 6 seats per row — 3 seats to either side of the aisle — are used to size the cabin bays. The distribution of cabin bays is required to have a gross length equal to the cabin length of an A321, while each cabin bay used for the BWB's is required to have a cabin width equal to the A321 cabin width. Furthermore, the A321 has a cargo hold volume of approximately 2048 ft^3; therefore, each BWB configuration is required to have an equal cargo holding volume, placed under the cabin bays.

Number of seats	220
Cabin length, ft.	146.0
Cabin width, ft.	12.14
Cabin height, ft.	6.99
Seat width, in.	18.0
Aisle width, in.	19.0
Armrest to armrest width, in.	143.0
Cargo hold volume, $ft.^3$	2048.3
Long range cruise altitude, ft.	35,000
Maximum range, nmi	3000
Cruise Mach number	0.78
Maximum payload, lb.	56,000

Table 3.7: Cabin dimensions and mission characteristics for the A321 used as the baseline configuration for the example trade study (values taken from [1]).

Given the required cabin and cargo bay dimensions, four BWB configurations are developed using the BWB design tool and a full system simulation is conducted on each design. The configurations had 1, 2, 3, and 4 cabin bays, respectively, with the cargo bay placed under the centerbody cabin bay for each configuration. The distribution of the cabin bays throughout the centerbody are chosen to closely resemble the arrangements used in studies conducted by Boeing [**?**], with the aft edges of the cabin bays aligned. The planform geometries of each of the configurations, as well as the distribution of the cabin bays, are shown in Figure 3-4. Table 3.8 gives the resulting planform areas and wingspan for each configuration.

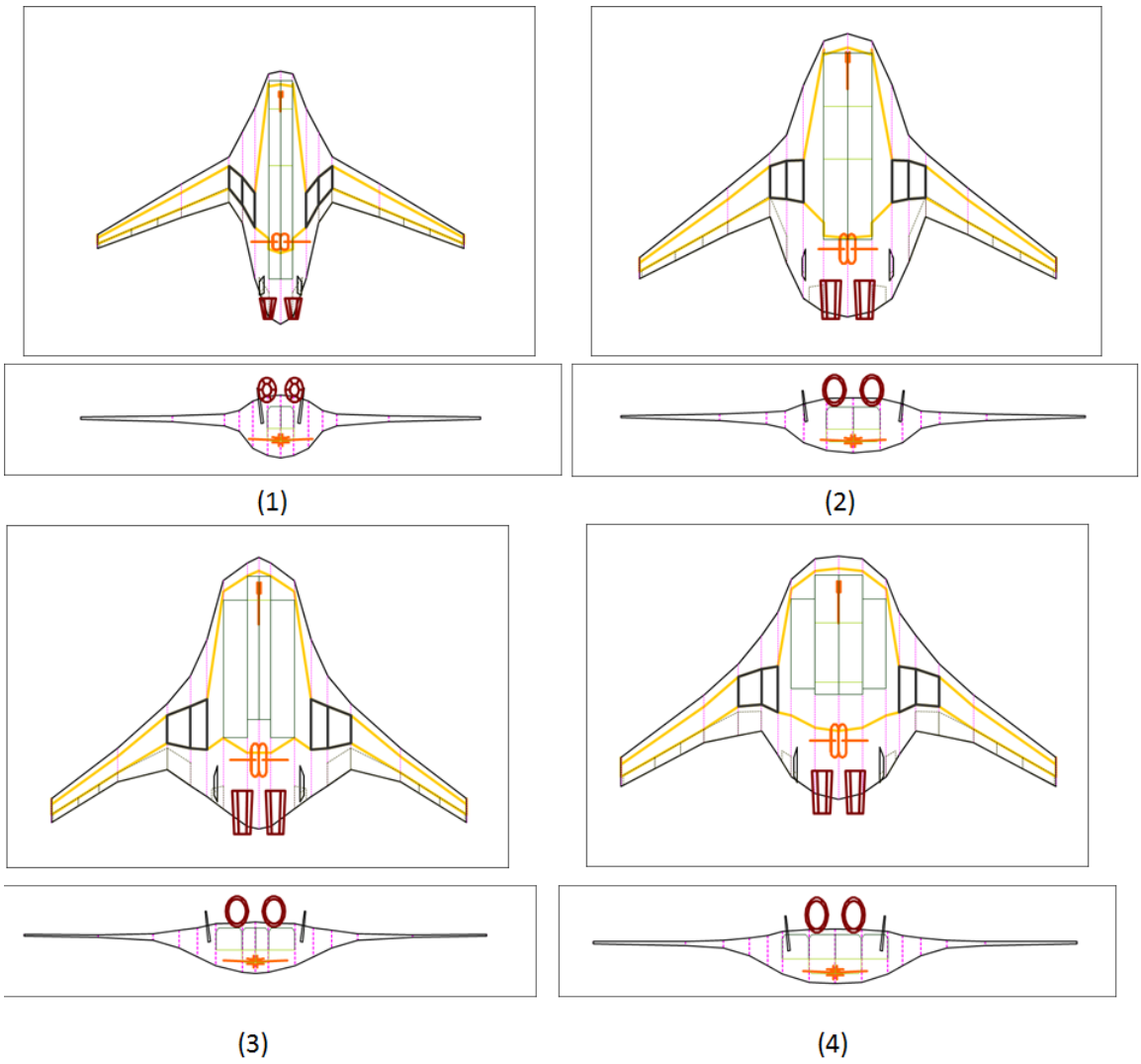


Figure 3-4: Planform geometry and cabin arrangement for each of the cabin bay configurations.

Discussion of Results

Table 3.9 shows the performance quantities of interest calculated by the BWB design tool for each of the studied configurations. From the table, it can be seen that the 1-cabin and 2-cabin configurations are not statically balanced, although the other performance metrics of interest for the two configurations are quite promising. Several other planform geometries were analyzed for each of the two cabin choices, but all were found to be statically unstable. The placement of the engines on the aftmost part of the aircraft play a significant role in the CG of each configuration falling well behind

Config.	# cabin bays	Planform area, ft.^{2}	Wingspan, ft.
1	1	8869	181
2	2	8284	206
3	3	7312	209
4	4	7791	219

Table 3.8: Geometric parameters for each of the configurations.

the calculated neutral points, along with the placement of landing gear, which was constrained by the placement of the lower carbo bay. These problems highlight the arduous task that a design configurator faces in balancing an aircraft configuration, which the BWB design tool simplifies. The large negative static margin for the 1-cabin configuration highlights that the particular choice of engine placement may not be practical for such a configuration and should instead follow the more traditional approach of placing the engines under the wings. Clearly, this choice of engine location does not allow the boundary layer ingestion that the SAI aircrafts relied upon to help with fuel burn, nor does it shield noise, which is one of the BWB's most promising features. Similar problems exist for the 2-cabin configuration, although a statically stable configuration seems possible. However, given the increased fuel burn calculated for the 2-cabin configuration, this configuration seems inferior to the 3- and 4-cabin configurations.

This study shows that, for the four cabin arrangements studied, the configurations which most resemble current BWB designs provide the best performance for the given mission. That is, the 3- and 4-cabin bay arrangements have decreased centerbody chords relative to conventional configurations and a resultingly decreased planform area. Of the four aircraft geometries analyzed in this trade study, the 3-cabin configuration is clearly the superior configuration, since it is statically stable, has a relatively low OEW, and a relatively low fuel burn. Using this study, the design configurator could perform a follow-up study on the two most promising cabin bay arrangements — the 3- and 4-bay configurations — by studying different planform geometries for each arragnement. Such a study could easily be performed in the BWB design tool.

Config.	Static margin (beg. cruise), %	OEW, lb.	Fuel burn, lb.	Cruise $M\frac{L}{D}$
1	-32.6	201,560	45,717	19.04
2	-13.1	202,224	53,335	18.16
3	3.5	201,337	46,940	18.66
4	4.6	213,079	50,048	18.34

Table 3.9: Performance quantities of interest calculated for each of the configurations.

Chapter 4

A Framework for Managing Complexity in Trade Space Exploration

At the conceptual design stage of a typical engineering system, a design configurator must explore the effects of a number of design variables to gain an understanding of the response of system quantities of interest to changes in the values of design variables. This process is typically performed via some trade-space exploration technique. For complex engineering systems, the number of design variables can be on the order of thousands, perhaps even infinite for continuous shape optimization, rendering an effective exploration of the full design space practically impossible. Thus, designers circumvent this problem by either narrowing their exploration to a small region of the design space — thus limiting the ranges of each design variable — or, as is most often done, fixing a number of the design variables at designer-chosen values, thereby reducing the dimension of the design space. Clearly, neither approach provides a robust framework for effective design space exploration, since they rely on "good" *a priori* choices of range restrictions for the former methodology and parameter values for the latter.

This chapter primarily concerns the development of a framework which addresses these problems by utilizing common hierarchies in engineering systems. This frame-

work can be utilized either to reduce the dimension of the design space or to enhance the quality of the design space. By partitioning the full-dimensional design vector along discipline lines - i.e. according to these natural hierarchies - the designer can perform trade studies using design variables which are intuitive to manipulate, while simultaneously ensuring that parameters are set to "good" values for each sample generated in the design space. Alternatively, parameters can be introduced into the design space and subsequently optimized for each sampled design to provide a better representation of the space. The first half of this chapter describes the motivation for the framework, along with its mathematical formulation and a general process flow. The latter portion of this chapter describes the framework through an example problem. The BWB configuration chosen for the Silent Aircraft Initiative is used as a baseline design to which the framework is applied. Application of the framework on this problem is shown to reduce the system objective of each sampled design by an average of 1% relative to the original exploration using only four total design variables.

4.1 A Partitioning Approach Based on System Hierarchies

The ability to decompose the design of most engineering systems to conform to distributed MDO architectures points to an inherent hierarchical structure of such systems. That is, the design variables which describe many engineering systems can often be partitioned into sets based on the level of fidelity of the iteration in the design process. For example, in conceptual design, high-level geometric variables are often used to analyze the engineering system, while lower-level design variables are either ignored or fixed as parameters in the design space. At a later iteration of the design process, perhaps during preliminary design, more design variables are used to describe the system, many of which were placed in the lower-level set of the conceptual design phase.

By utilizing these hierarchies in design variables describing an engineering system, a natural partitioning of the design space follows. That is, the design space used for a particular design phase can be decomposed into an upper- and lower-level set of design variables. Therefore, upon decomposition, the design space becomes bi-level. In particular, the design variables over which the designer chooses to conduct the trade study belong to an upper-level set x_u, while the lower-level set x_ℓ is comprised of the design variables which have subtle effects on system performance or which are too numerous to effectively explore. Often, the lower-level design variables can be considered discipline-level variables in the context of a multidisciplinary system, in that these variables primarily affect a single discipline in the system and are exclusively controlled by a discipline expert. Figure 4-1 gives an illustration of the partitioning process. Specifically, the full-dimensional design vector is partitioned into two lower-dimensional vectors, where the upper-level design vector defines the new design space for the design of experiments (DOE) and presumably represents a large reduction in dimension relative to the original design vector. The lower-level design vector is then comprised of the design variables from the original design vector which were not placed in the upper-level set.

Table 4.1 shows potential partitionings of the design space during conceptual design for several engineering systems. The particular choice of the design vector partition is left to the designer. By relying on the human designer, the intuition and expertise is explicitly placed in the design loop.

Engineering system	Upper-level variables	Lower-level variables
Aircraft design	chord, tail volume,	landing gear location
	sweep, aspect ratio	spar location, airfoil shape
Rocket design	length, diameter	trajectory, fairing shape
	static thrust, # stages	interstage height
Chemical process design	epuipment shapes	control variables
Turbofan engine design	number of stages, T_{t_4}	combustor shape, nozzle shape
	bypass ratio	compressor blade shape

Table 4.1: Examples of potential partitionings in engineering systems for conceptual design.

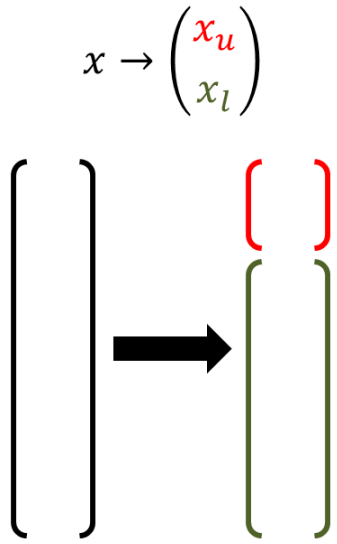


Figure 4-1: Illustration of paritioning the design vector. The design vector is decomposed to a low-dimensional upper-level vector (red) and a higher-dimensional lower-level vector (green).

4.2 A Bi-Level Framework for Managing Complexity based on Partitioning

During conceptual design, a designer is primarily concerned with exploring large regions of the design space, since it is often necessary to analyze a wide array of designs early in the design process. This is generally accomplished by choosing a set of design variables and performing a particular DOE, which is a collection of statistical techniques which provide a systematic methodology for sampling the design space [7]. Such methods are generally useful for studying the effects of the design variables on the output parameters of interest. Therefore, these exploration techniques are often used in identifying driving characteristics of the system, as well as appropriate design variable ranges, to aid in the setup of a formal optimization problem. Thus, because the success of an engineering system optimization hinges on the problem formulation, the design space exploration phase of the design process is a critical step in designing a system which performs well.

4.2.1 Mathematical Formulation of Framework

Given the conformance of many engineering systems to the hierarchical model described above, it is straightforward to utilize this characteristic in exploring the design space and performing a DOE. Moreover, because the designer often plays an integral role in this phase of the design process, this natural reliance on the human designer can be utilized. Once the design vector is appropriately partitioned, the designer is left with a set of upper-level design variables which can be used for a DOE. However, the values of the lower-level variables within the lower-level design vector must be chosen. As previously discussed, a designer would generally choose to fix these parameters to some baseline values. However, keeping the lower-level variables constant for each upper-level sample may not be truly representative of the design space, since regions of the design space may be quite sensitive to parameter choices. To alleviate this potential problem, the bi-level framework uses a suitable lower-level objective function $g(\cdot,\cdot)$, chosen by the designer, to optimize the lower-level variables for each upper-level sample. The general mathematical formulation for this procedure is given in Equation 4.1.

$$\begin{aligned} \underset{x_u}{\text{Explore}}\ & f(x_u; x_\ell^*) \\ \text{s.t. }\ & c(x_u; x_\ell^*) \geq 0 \\ & x_\ell^* = \arg\min_{x_\ell} \{ g(x_\ell; x_u) \mid c_\ell(x_\ell; x_u) \geq 0 \}. \end{aligned} \tag{4.1}$$

where f is the system objective, c is the system level constraint set, g is the lower-level objective function, and c_ℓ is the lower-level constraint set.

Equation 4.1 resembles the general bi-level programming problem formulation, except that the upper-level objective function f is not optimized; therefore, the framework circumvents many of the problems due to nonconvexity and any complications of the optimality conditions which commonly arise in bi-level programming problems [10]. Instead, a new notation is introduced, wherein the usual "$\min_{x_u}$" term is re-

placed by the "Explore $_{x_u}$" term. Because the bi-level framework is developed for the design space exploration phase of the design process, it is unnecessary for the designer to find the best design for the given requirements, hence the minimization statement of the general bi-level optimization problem is unnecessary. Rather, the designer is most interested in analyzing the system for a range of upper-level design vectors, hence the adoption of the "Explore" notation in the above formulation. Thus, this notation can be interpreted as the upper-level variables x_u being used as factors (in the DOE context), while observations of the system objective f are taken across the design space.

Note that the upper-level objective is a function of only the upper-level design variables, with the optimal values of the lower-level variables specified as parameters; alternatively, the lower-level objective is a function of the lower-level design variables, with the upper-level variables fixed as parameters. Figure 4-2 illustrates the bi-level framework. In this figure, a surrogate model generated from samples of the upper-level design variables is constructed, where each sample has had its corresponding lower-level design variables optimized with respect to a lower-level objective function. This process ensures that many of the parameters describing the system are set to good values during the design space exploration while simultaneously reducing the dimension of the original design space. A general process flow for the bi-level framework is described in the following section.

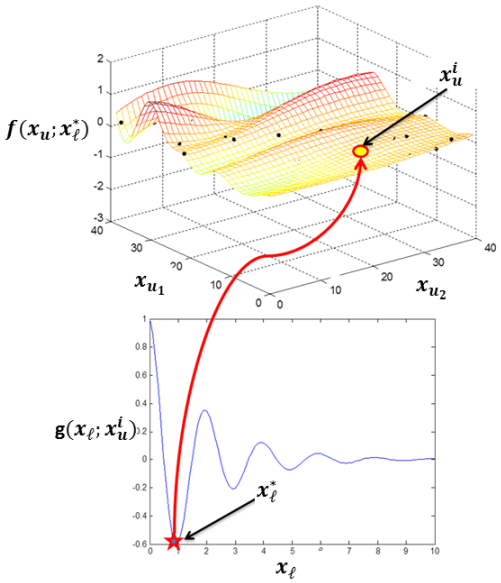


Figure 4-2: Illustration of the bi-level framework. Each upper-level sample used to generate the system surrogate has had its lower-level design variables optimized with respect to a lower-level objective function.

4.2.2 Process Flow of the Framework

Figure 4-3 shows the Extended Design Structure Matrix for the bi-level framework [21]. The designer serves to initiate the process by providing the design variable partitioning and subsequent lower-level optimization problem. Furthermore, the optimization routine is general, in that any optimization procedure can be used to solve the lower-level problems. A feedback loop is placed between the first and last steps in the process so that the designer can guide the DOE by increasing the sample size or changing the lower-level objective. The bi-level framework is comprised of four primary steps, as shown in Algorithm 1. The following sections provide a detailed discussion of each of the steps.

Algorithm 1 Bi-level framework for trade space exploration

1. Partition the design vector into upper- and lower-level sets; choose design variable bounds and lower-level objective function.

2. Generate N samples across the upper-level design space.

3. For each upper-level sample, conduct optimization of lower-level objective function over lower-level design variables until convergence criteria are satisfied.

4. Evaluate upper-level objective function for each of the sampled upper-level design variables and corresponding optimal lower-level variables. Post-process results, e.g., construct a surrogate model and optimize, visualize design space behavior, or perform a sensitivity analysis.

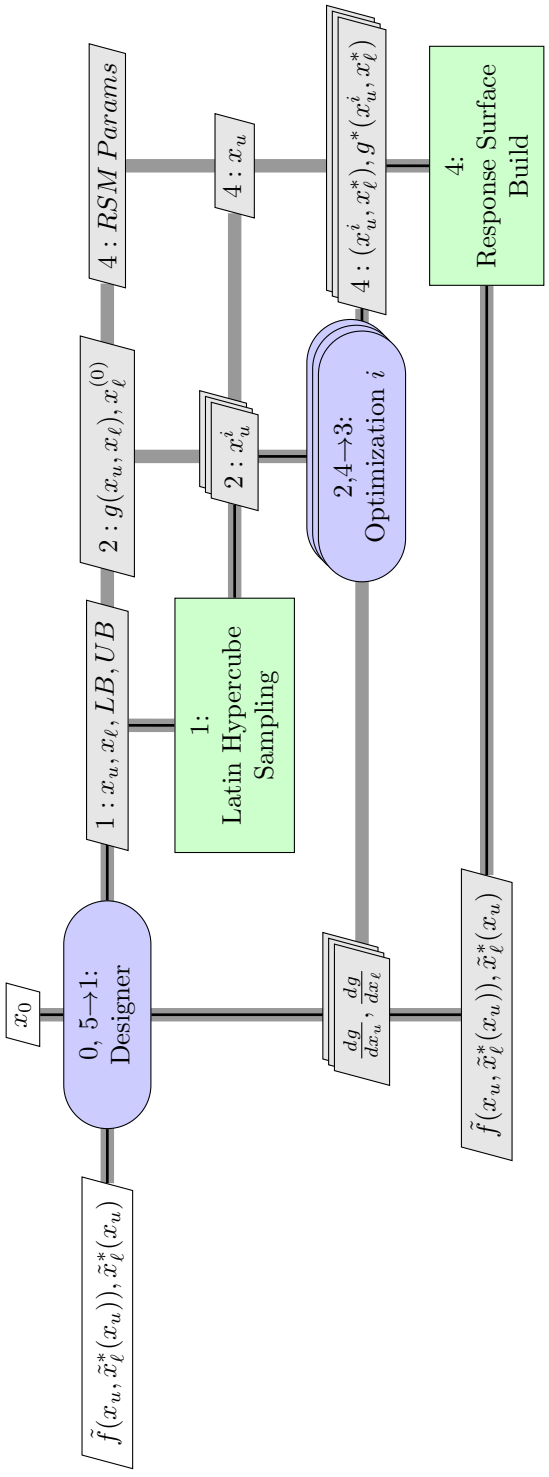


Figure 4-3: Extended Design Structure Matrix of framework process flow.

Step 1: Partitioning

Assuming the design problem is known and the design variables and system parameters have been chosen, the designer partitions the design variables into an upper- and lower-level set, as discussed in the preceding section. In particular, the upper-level variables, x_u, are chosen to be high-level, or system-level, variables which are key drivers in system performance. Alternatively, the lower-level variables, x_ℓ, are chosen to be discipline-level design variables which effect system performance, but which are often too numerous to effectively explore.

The bi-level framework does not automate the partitioning process. Rather, the designer is relied upon to choose the most appropriate decomposition. This reliance on the human designer is primarily motivated by the general lack of designer utilization in the design process. However, automated techniques are certainly possible in the form of heuristic methods.

Step 2: Upper-level Sampling

Depending on the experiment chosen for the trade study, appropriate samples of the upper-level design variables must be generated. The examples used throughout this thesis generate samples via Latin hypercube sampling, whereby the design space is divided into ℓ divisions for each design variable and the levels are then randomly combined such that each level of each variable is only used once [31]. The goal of generating samples for design space exploration is generally to provide a large coverage of the full design space allowing for a representative model of the true design space. Typically, the larger the dimension of the design space — that is, the more design variables over which the designer must explore — the more samples are required to represent the space. Thus, because computationally expensive simulations are often necessary to evaluate system performance, large design spaces may become computationally intractable. The design vector partitioning helps alleviate this problem.

Step 3: Lower-level Optimization

Given the lower-level design variable set, a suitable lower-level objective function g is chosen for the lower-level optimization, along with a lower-level constaint set. The general formulation of the lower-level problem allows any optimization method to be utilized to find the optimal lower-level values. This is particularly useful in engineering design since many of the analyses rely on blackbox codes which are not amenable to gradient-based approaches. Therefore, a gradient-free method may be used for legacy codes which cannot easily compute a derivative, or a mixed-integer approach can be used when discrete variables are used in the lower-level set, or a multifidelity method can be used to speed up the lower-level optimization. Once an optimization procedure is chosen, the lower-level variables are optimized for each sample generated in the previous step. Thus, a total of ℓ optimizations must be performed.

The lower-level objective function will generally be chosen to be some discipline-specific quantity of interest which affects the upper-level objective function. Furthermore, the lower-level objective function will be, in general, easier to optimize with respect to than the system objective, since it represents only a subset of the full system analysis. This is especially useful in the context of the bi-level framework due to the potentially large number of requisite lower-level optimizations. Moreover, because each of the optimizations is independent, the lower-level problems can be parallelized, which could greatly reduce the time required to complete this step.

Step 4: Upper-level Objective Evaluation and Post-Processing

Properly designed experiments are essential for effective computer utilization [37]. An application for performing DOE methods is to construct approximations of the computationally expensive simulation and analysis codes to provide surrogate models that are sufficiently accurate to replace the original code. These surrogate models provide a means to construct simple and fast approximations of complex computer analyses, greatly speeding up the design process. Figure 4-4 gives an illustration of

the process of sampling a design space and constructing a fast approximation model of the blackbox simulation.

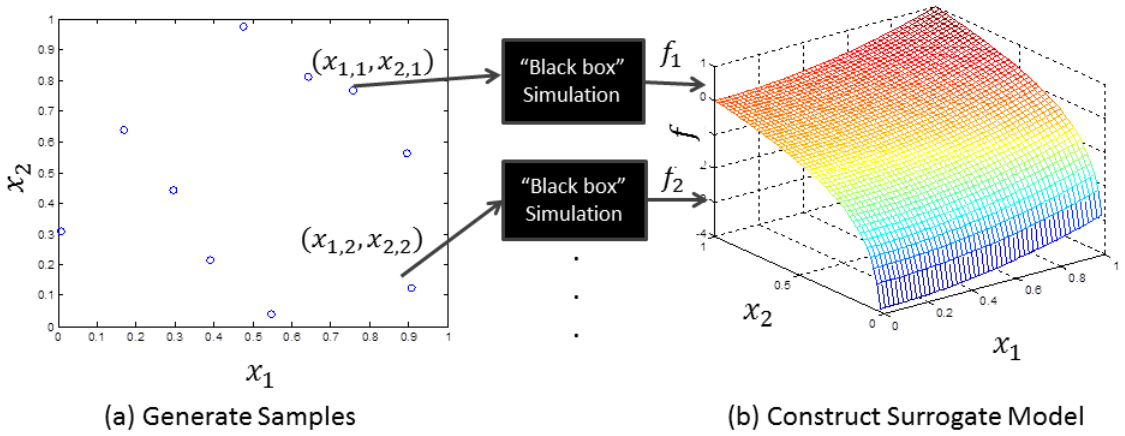


Figure 4-4: Illustration of the process of the design and analysis of computer experiments.

Once a lower-level optimization has been performed for each of the samples generated for the DOE, the system objective — that is, the upper-level objective function — is evaluated for each of the samples. This involves performing a system simulation for each of the samples and any subsequent post-processing. For example, the designer may want to use the samples to build a surrogate model of the objective function, which can be used for a system optimization. Alternatively, a surrogate model of the optimal lower level variables may be constructed as a means to better predict values of parameters of the system. The potentially small dimension of the design space allow a means to calculate all of the main effects and interacting effects for each of the upper-level design variables, which provide significant insight into the design space.

4.3 An Example Problem Utilizing the Framework: Redesign of the SAX-40 Aircraft

To illustrate the general process a designer would follow in applying the bi-level framework to a trade study, an example problem is used which involves the exploration over BWB designs for fuel burn. Consider the design of the SAX-40 aircraft, as discussed in Chapter 3. The SAI analyses conducted an optimization of this configuration which used 5 design variables describing the planform shape of the aircraft. These design variables represented the outerwing section chord, a kink section chord, the outerwing sweep, the outerwing span, and the x-location of the spanwise kink location.

For the original SAX-40 design, a single, offline optimization of the airfoil shape was performed and used throughout the planform optimization process; therefore, the airfoil shape was a parameter throughout the planform optimization. However, because the optimal airfoil shape is a function of the planform geometry, as well as the cruise parameters, the airfoil shape used for the SAX-40 design was not necessarily optimal. In applying the bi-level framework to this problem, the planform geometry is still explored over for effects on fuel burn, but the airfoil shape is introduced into the design space as a lower-level variable set. A surrogate model is built over the outerwing planform shape variables both with and without application of the bi-level framework. Results show that application of the bi-level framework on the SAX-40 design exploration problem reduces the average fuel burn of the explored configurations by roughly 1% relative to the original exploration, all while using only four total design variables.

4.3.1 Design Space Partitioning and Problem Setup

In order to simplify the example problem and as a means to visualize the entire design space, two variables are chosen as the upper-level design variables, while the other five design variables of the original SAI problem are fixed to the values used for the SAX-40 configuration. Specifically, the outerwing section chord and the outerwing

sweep are chosen as the upper-level design variables, since these variables have a large effect on both fuel burn and cruise drag.

The airfoil shape is defined by 5 angles placed at control points along the airfoil surface which are interpolated using Bezier curves, as described in [20]. Figure 4-5 shows the airfoil parameterization used to define the airfoils. Given this parameterization of the airfoil shape, a natural selection of the lower-level design variables is to use a subset of these angles. In particular, the two angles at controls points 2 and 4 in the figure are chosen as the lower-level design variables, since these two angles most affect the airfoil camber.

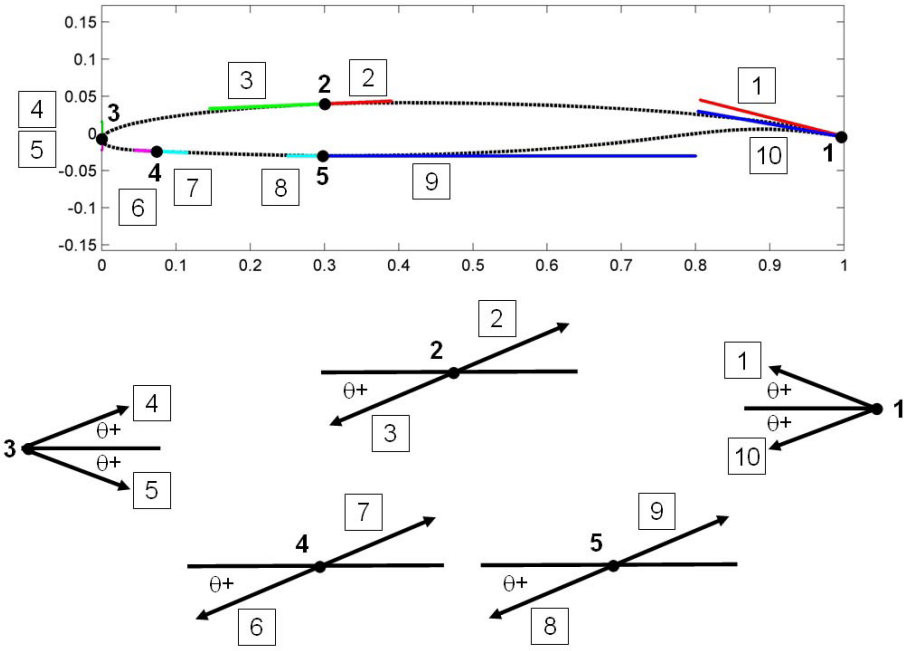


Figure 4-5: Airfoil parameterization used for the lower-level optimization (taken from [20]).

The system objective is chosen to be fuel burn; therefore, a natural choice for the lower-level objective function is the cruise drag over the outer airfoil. Therefore, the mathematical formulation of the example problem is given in Equation 4.2.

$$\begin{aligned} &\underset{c,\Lambda}{\text{Explore}}\ W_f(c,\Lambda;\theta_2^*,\theta_4^*) \\ &\quad \text{s.t. } h(c,\Lambda;\theta_2^*,\theta_4^*) \leq 0 \\ &\qquad \theta_2^*,\theta_4^* = \underset{\theta_2,\theta_4}{\arg\min}\{C_D(\theta_2,\theta_4;c,\Lambda) \mid h_\ell(\theta_2,\theta_4;\Lambda,c) \leq 0\} \end{aligned} \tag{4.2}$$

where W_f is the total fuel burn, C_D represents the cruise drag, c is the outerwing section chord, Λ is the outerwing sweep, θ_2 is the angle at control point 2 on the outerwing airfoil, and θ_4 is the angle at control point 4 on the outerwing airfoil.

4.3.2 Lower-Level Optimization Problem

Each 2-D airfoil section is analyzed using MSES such that the viscous, profile, and wave drag data are generated over a range of C_L values. Due to the computational time required to generate such data using MSES, it is too computationally expensive to include the MSES simulations within the optimization loop. Therefore, the MSES data is generated offline over a range of sweep angles and airfoil shapes and appropriate drag surrogates are built. Figure 4-6 shows the surrogates constructed for the various drag components using these offline runs. The specific surrogate models constructed for this problem are Kriging models using a Gaussian correlation function. The DACE toolbox is used to construct the Kriging surrogates [23]. The lower-level optimization problem, and thus application of the bi-level framework, is then able to proceed.

4.3.3 Results

A total of 40 upper-level samples are generated for this problem using a latin-hypercube sampling technique. For each sample, a system simulation is performed using both the original outerwing airfoil and the airfoil which provides minimum cruise drag for the configuration using the lower-level surrogate model. A Kriging surrogate model is constructed for each case. Figure 4-7 shows the surrogate mod-

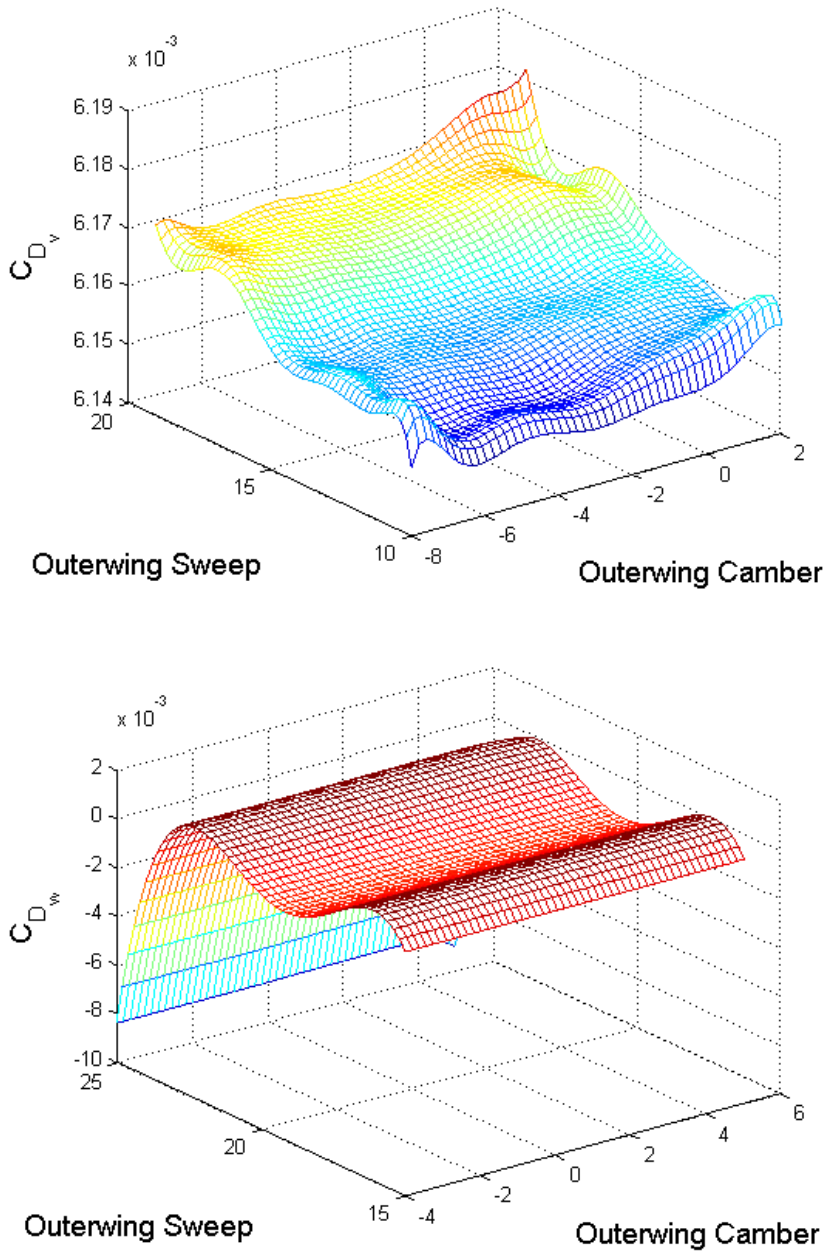
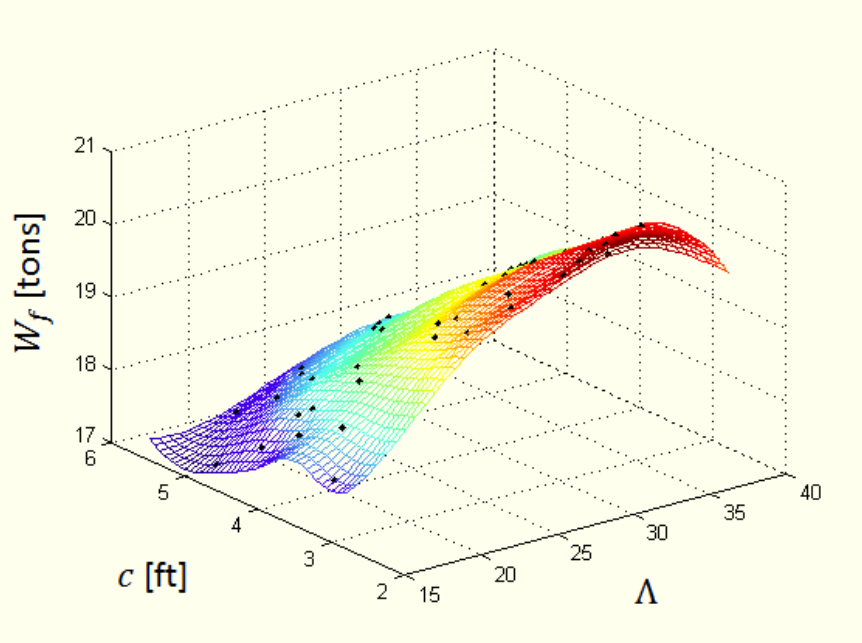


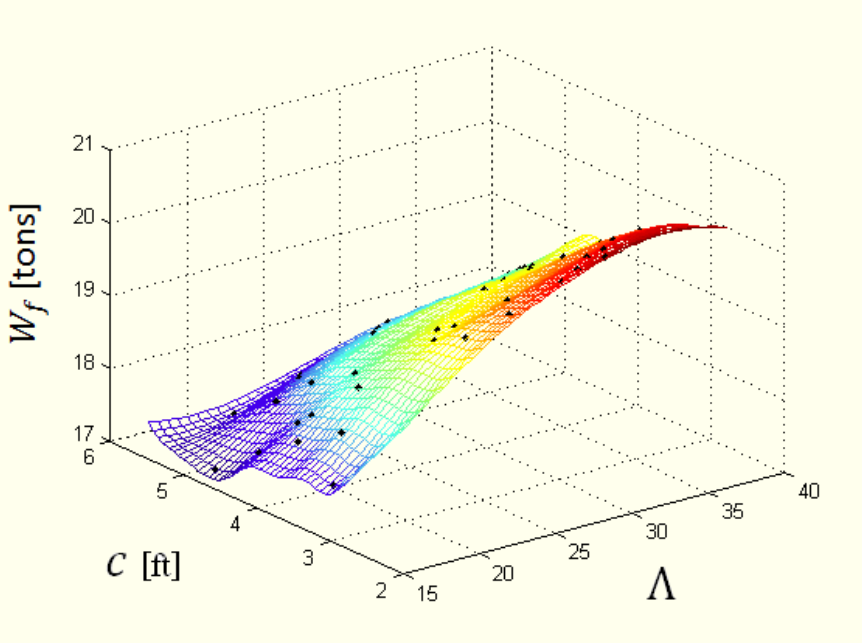
Figure 4-6: Kriging surrogate models generated for a specific C_L for (top) the viscous drag and (bottom) the wave drag used for the lower-level optimization over the chosen design variables.

els generated from this study. Given the small design space used for this example problem, it is no surprise that the two surrogates are quite similar. A qualitative comparison of the two surrogate models shows that the bi-level framework serves to shift the surrogate down and smooth the model compared to the default case, thereby providing a better design space over which to explore. Furthermore, the surrogate constructed after applying the bi-level framework provides an average decrease in fuel burn of roughly 400 pounds, or roughly 1%, relative to the surrogate constructed with the default outerwing airfoil. While seemingly insignificant, a reduction in fuel burn of 400 pounds can be extrapolated over the lifetime of a single aircraft. For example, if the aircraft is flown for 10 to 12 hours per day, at a fuel cost of $3 per gallon, a 1% reduction in fuel burn can add up to millions of dollars. Thus, application of the bi-level framework on this problem shows that the design space can be enriched by introducing very few parameters as lower-level variables, which serves to construct a better surrogate model, and may provide significant improvements for the chosen system configuration.

While the 1% reduction in fuel burn provided by the bi-level framework is likely within the accuracy of the SAI codes, application of the framework still highlights one of its major advantages; namely, the design space is reduced to include only two design variables, thereby allowing the designer to visualize the entire space. This is particularly useful for trade studies, since it eliminates the need to take cuts of the design space (by fixing those design variables which are not being visualized) and thus provides a holistic view of the design variable's effects on system performance. In fact, a primary goal of many Visual Design Steering methodologies is the reduction of the design space to include as few design variables as possible, thus allowing the designer significant insight into the design problem through visualization of the space [47]. The bi-level framework accomplishes this while simultaneously ensuring that the values of the parameters describing the system are chosen as good values relative to some discipline-level metric.



(a)



(b)

Figure 4-7: Kriging surrogate models generated using (a) a default outerwing airfoil section and (b) a cruise-drag optimized outerwing airfoil section for each sample. The RMS difference between the grid points of the surfaces is about 400 lbm., representing an approximately 1% reduction in estimated fuel burn.

Chapter 5

Demonstration of the Framework on a BWB Design Problem

The BWB design tool is developed as a means to perform multidisciplinary trade space studies for the conceptual design of a BWB aircraft. Chapter 3 shows several example trade studies which may be performed using the BWB design tool. Furthermore, Chapter 4 develops a framework for decomposing the original system design problem so that effective trade studies may be performed. This chapter applies the bi-level framework on two BWB trade studies using the BWB design tool. Results show that the bi-level framework can be utilized to either enrich the design space by optimizing over parameters or it can simplify the design space through an appropriate partitioning and lower-level optimization. Both example problems explore the effects of BWB planform geometry changes by modifying the two outermost spanwise sections; thus, the centerbody is fixed to a design similar to the SAX-40 configuration, while the outerwing is modified. The first example problem highlights the bi-level frameworks ability to enrich the design space by considering the cruise Mach number as a lower-level design variable and optimized for the cruise ML/D for each sampled configuration. The second example problem applies the framework to a flight planning problem by exploring over the flight operating costs (FOC) for three missions by optimizing the flight profiles for each of the three missions. Results show that application of the bi-level framework to each of the example problems leads to the ex-

ploration over designs which have up to a 4% improvement in performance objectives compared to the default case.

5.1 Cruise Speed as a Lower-level Variable

The cruise speed of a configuration is often fixed to a default value — specified by either a system requirement or from some baseline choice — when a designer is exploring over planform geometries. This seems to simplify the trade study for a designer by providing common baseline flight characteristics for which to compare the different configurations. However, fixing the cruise Mach number may be ineffective in gaining intuition into the effects of planform geometry changes on aircraft performance, since different BWB configurations can have different values for the optimal cruise Mach number. The fuel burn of a configuration is quite sensitive to the cruise speed, since the ML/D term is a significant driver in the Breguet range equation. Therefore, this problem explores application of the bi-level framework to a design space exploration over the outerwing geometry, as in the example trade study in Chapter 3, by considering the cruise Mach number as a lower-level design variable; this study uses the BWB design tool to perform the system simulations. Application of the bi-level framework results in an average reduction in fuel burn of roughly 800 lbm (4%) relative to a default value for the cruise speed.

5.1.1 Problem Setup

This problem investigates using the bi-level framework to explore over BWB planform geometries while considering the cruise speed (i.e., the cruise Mach number M_∞) as a lower-level variable. The upper-level variables chosen for this problem correspond to the width, chord, and sweep of the two outermost spanwise sections as parametrized in the BWB design tool; therefore, there are a total of 6 upper-level variables. Furthermore, modification of these two sections has the effect of changing the outerwing shape, so this problem is effectively an exploration over outerwing shapes. The centerbody of the BWB geometries considered in this study is fixed to the one used for

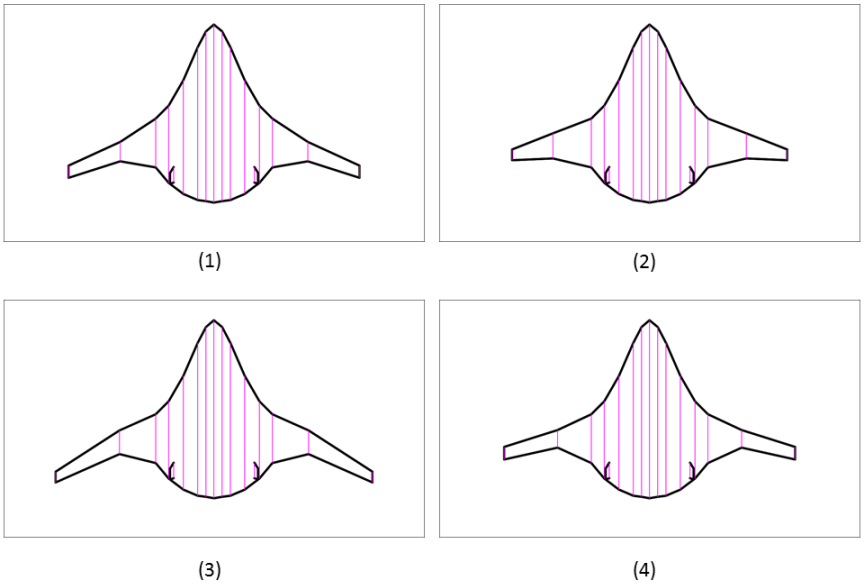


Figure 5-1: Four of the planform geometries generated by the sampling procedure.

the SAX-40 study in Chapter 3. Figure 5-1 shows four of the planform geometries generated for the problem by a Latin-hypercube sampling procedure.

The optimization problem considered is,

$$\begin{aligned} &\underset{c,b,\Lambda}{\text{Explore}} \; W_f(c, b, \Lambda; M_\infty^*) \\ &\quad \text{s.t.} \; h(c, b, \Lambda; M_\infty^*) \leq 0 \\ &\qquad M_\infty^* = \underset{M_\infty}{\arg\max}\{ML/D(M_\infty; c, b, \Lambda) \mid h_\ell(M_\infty; c, b, \Lambda) \leq 0\} \end{aligned} \tag{5.1}$$

where W_f is the weight of the fuel burned for the specified mission, c represents the outerwing chord, b is the outerwing span, Λ is the outerwing sweep, and M_∞^* is the optimal cruise speed computed by the lower-level optimization for the specified configuration.

5.1.2 Cruise Speed Optimization Procedure

To find the optimal cruise Mach number for each BWB configuration, a full system simulation is run for a range of Mach numbers (between 0.7 and 0.9) and a surrogate model is fit to the data. Once the surrogate is constructed, an active set algorithm is implemented to find the cruise Mach number which provides the largest ML/D value for the given aircraft configuration [30]. Figure 5-2 shows two such example surrogates constructed for two of the configurations samples for the study. Each configuration corresponds to a different optimal cruise Mach number, as expected.

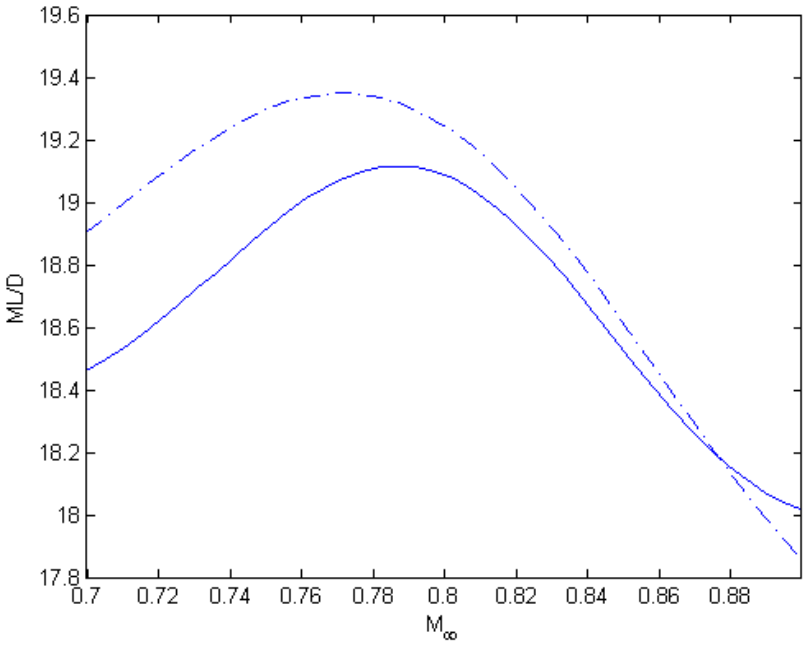


Figure 5-2: Surrogate models of M_∞ vs ML/D generated for two different configurations used in the problem. These functions were used to find the optimal cruise speed for each of the configurations for application in the bi-level framework.

5.1.3 Results

A total of 40 samples are generated across the design space. Each sample describes a different configuration in the BWB design tool. A cruise Mach number of 0.80 is used as the default value for the simulations not employing the bi-level framework. Figure 5-3 shows the results of the lower-level optimization for each of the sampled

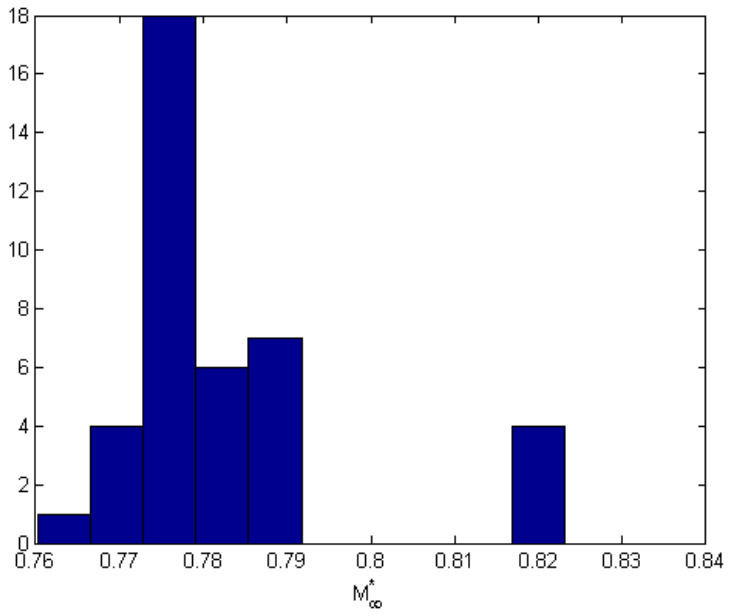


Figure 5-3: Histogram of the optimal cruise Mach numbers found for the sampled configurations.

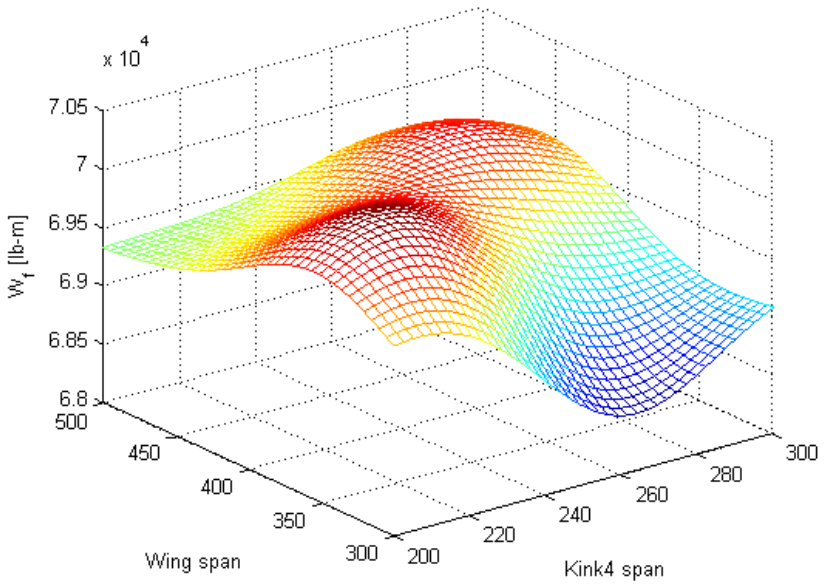
configurations used throughout the study. From the histogram, it is clear that the optimal cruise speed does not match the default value of $M_\infty = 0.8$; for most configurations, the optimal cruise Mach number is around 0.78. It is around this Mach number that the wave drag on the configuration becomes insignificant — and thus the L/D value becomes largest — while still maintaining a relatively large cruise Mach number, since this value scales the L/D term in the lower-level objective function.

Figure 5-4 shows the upper-level surrogate model constructed for the problem for both the fixed cruise speed case and after application of the bi-level framework for a slice of the design space (since the upper-level space contains 6 total design variables). The qualitative differences between the two surfaces is obvious upon first inspection, since the surrogate constructed using the optimal cruise speed exhibits a clear reduction in W_f relative to the default case throughout the majority of the shown design space. Thus, significant differences in the shape of the design space may exist when parameters are not chosen to be the "best" values. Specifically, this example shows that slight changes in the values of parameters which may drive the system objective can lead to drastic differences in the design space; therefore, the designer

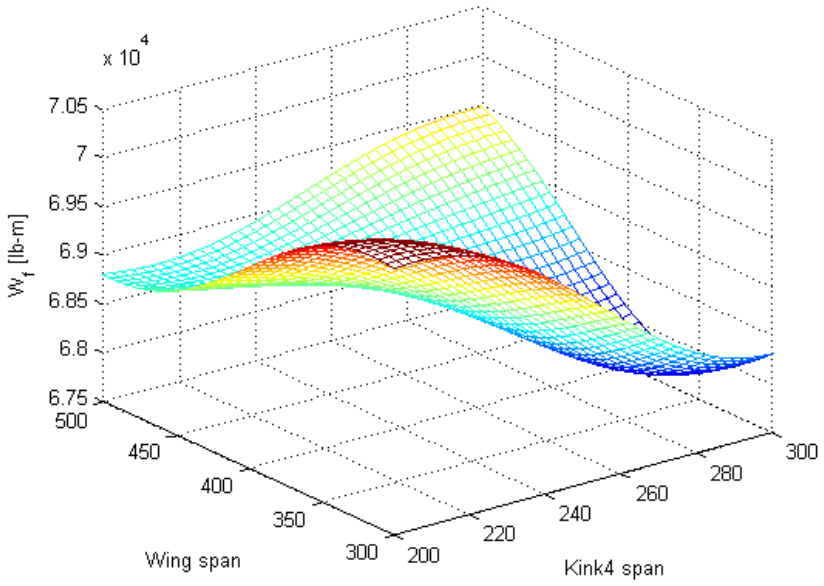
may be exploring over an inferorior design space during conceptual design, which could propagate throughout the design process and ultimately lead to a nonoptimal choice for the final design. Quantitatively, there is a root-mean square difference between the grid points of the two surrogate models of approximately 840 lbm, or a roughly 1.5% reduction in fuel burn for the optimal cruise speed. Following the discussion from the example problem of Chapter 4, a decrease in fuel burn of this magnitude can lead to substantial cost savings over the course of the aircraft's life. Thus, by optimizing a single lower-level variable, significant benefits may result from the utilization of the bi-level framework.

5.2 Mission Flight Profile as a Lower-level Variable

An important factor influencing the design of any aircraft is the cost to fly the aircraft over specific missions, or its flight operating cost (FOC). The FOC is generally a function of the fuel burned over the given mission, as well as the operational, maintenance, and crew costs for the time in which the aircraft is flown for the mission. Using the same upper-level choice as the preceding problem, the effects of planform geometry of a BWB configuration on a weighted FOC measure are investigated. For this problem, a total of two potential domestic missions are used to calculate a weighted FOC for an aircraft configuration. Each mission is characterized by its range; thus, mission 1 is a short-range 500 nautical mile flight similar to express flights between major cities, while mission 2 is a medium-range 1000 nautical mile flight corresponding to flights to airline hubs from major cities. Weights are chosen to correspond to the approximate frequency of each of the two flights per day. To apply the bi-level framework to this problem, the lower-level variables are selected to correspond to the flight trajectory for each of the missions. The BWB design tool allows the flight trajectory to be modified through a specification of the initial cruise altitude; thus, the lower-level variables are chosen to be the initial cruise altitude for each of the missions. Results of the



(a)



(b)

Figure 5-4: Kriging surrogate models generated using (a) a default cruise Mach number and (b) an optimized cruise drag Mach number (for ML/D) for each sample. The RMS difference between the grid points of the surfaces is roughly 840 lbm., representing an approximately 1.5% reduction in estimated fuel burn.

problem demonstrate that application of the bi-level framework leads to a nearly 4% improvement in weighted flight operating costs compared to the default case.

5.2.1 Problem Setup

As described above, the FOC is generally a function of the fuel burned during the flight, along with the time of the flight. The model used throughout this problem to calculate FOC for each of the missions is,

$$FOC = C_w W_f + C_t t \tag{5.2}$$

where W_f is the fuel burn,C_w denotes the price of fuel per gallon, t is the total time of flight from when the blocks are removed from the wheels to when the blocks are placed under the wheels (i.e. a block hour), and C_t is the price per block hour. C_t is taken from B737-500 historical data for Southwest Airlines which encapsulates the crew cost, maintenance cost, and ownership cost per block hour; the value of C_t used for this study is \$989 per block hour [4]. C_w is taken to be the six month average jet fuel price per gallon from September 2012 through March 2013 ; thus, the value of C_w used throughout this problem is \$3.07 per gallon [2]. A full system simulation using the BWB design tool provides both the fuel burn weight, as well as the time of the flight; therefore, calculation of the FOC using the tool is a straightforward task.

Since the objective of this problem is to explore the FOC for different BWB planform geometries, the same choice of upper-level variables as the first example problem is used. Therefore, the problem formulation using the bi-level framework is given in Equation 5.3.

$$\begin{aligned} &\underset{a_1,a_2}{\text{Explore}}\ 0.3FOC_1 + 0.7FOC_2 \\ &\quad \text{s.t. } h(c,b,\Lambda;a_1^*,a_2^*) \leq 0 \\ &\qquad a_i^* = \arg\min_{a_i}\{FOC_i(a_i;c,b,\Lambda) \mid h_\ell(a_i;c,b,\Lambda) \leq 0\} \end{aligned} \tag{5.3}$$

where a_i is the initial cruise altitude for mission i (which is used to describe the trajectory for the given mission), FOC_i is the flight operating cost for mission i. The weights placed on the FOC for each mission show that the greatest emphasis is placed on medium-range flights, since these flights seem to occur most frequently for airlines (about 70% of the two missions). Furthermore, from the formulation, it is clear that a total of two lower-level optimizations were required — corresponding to the two different missions — for each of the sampled configurations. The procedure for performing the lower-level optimizations is described in the following section.

5.2.2 Mission Profile Optimization Procedure

To compute the optimal initial cruise altitude for each of the three missions, a full system simulation is performed across a range of altitudes. From these simulations, a surrogate model for the FOC of the mission is constructed and subsequently optimized. Figure 5-5 shows an example surrogate constructed for the short-range mission.

Figure 5-5 shows that two local minima exist for this configuration, as is the case for the majority of configurations for each of the missions. Because of the existence of multiple optima, the initial point chosen for the optimization plays a key role in determining the initial cruise altitude to which the optimization converges. For the example surrogate shown, the local optimum is around 30,000 ft. and results in a smaller FOC than the local optimum near 36,000 ft. To ensure consistency amongst the lower-level optimizations, the same initial cruise altitude is used as the initial guess for a given mission, namely the default cruise altitude used for each mission. Table 5.2.2 gives the specified mission parameters for each of the two missions, as well as the bounds on the cruise altitude used for the optimization procedure.

Mission	Range, nmi	M_∞	Default cruise altitude, ft.	Cruise altitude range, ft.
1	500	0.82	35,000	30,000-40,000
2	1000	0.82	37,500	34,000-43,000

Table 5.1: Flight parameters for the 3 missions used in the example problem.

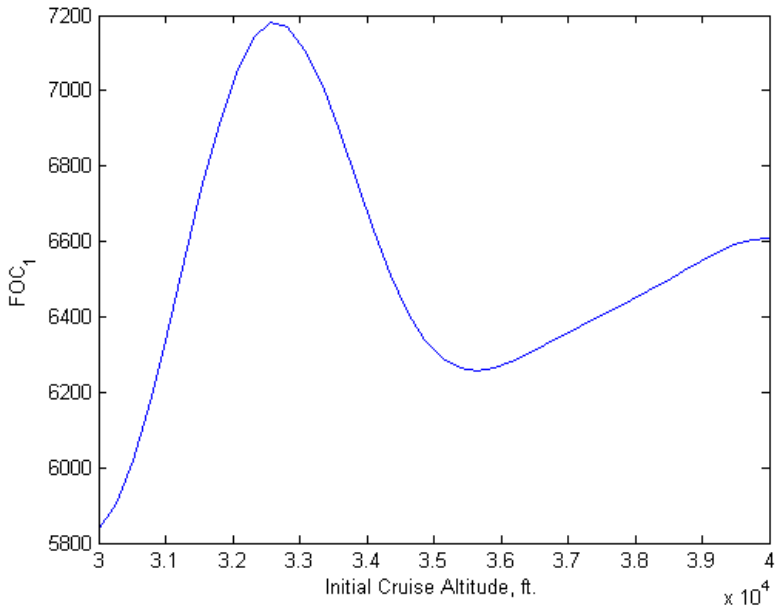
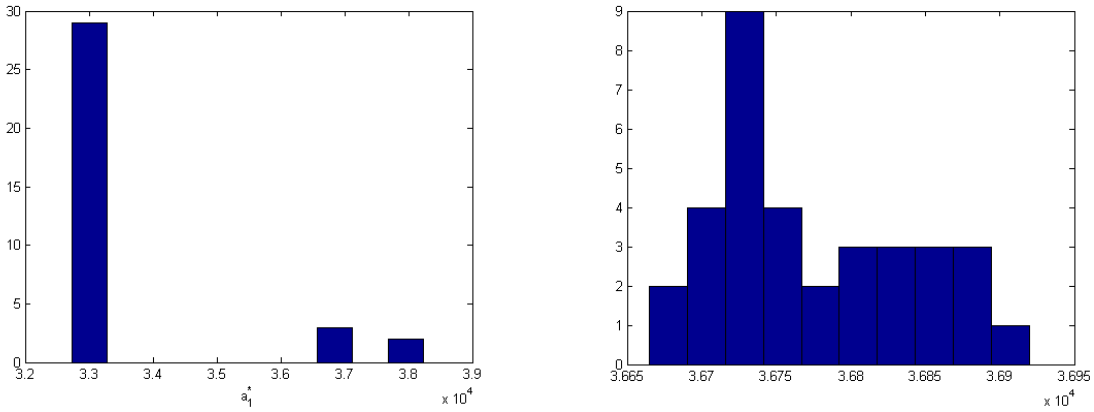


Figure 5-5: Example surrogate function generated for the flight operating cost as a function of the initial cruise altitude for the short-range mission. This surrogate was constructed for each sampled configuration in order to find the optimal initial cruise altitude.

5.2.3 Results

A total of 34 samples corresponding to different planform geometries are used for this problem. Figure 5-6 shows the results of the lower-level optimization for each of the missions. The optimal initial cruise altitude for each mission cluster around roughly the same altitude; specifically, the optimal short-range cruise altitude is around 33,000 ft., while the optimal cruise altitude for the medium-range mission is around 37,000 ft. The few outliers in optimal cruise altitude for the first mission are due to the local minima of the lower-level objective surrogate, as discussed above.

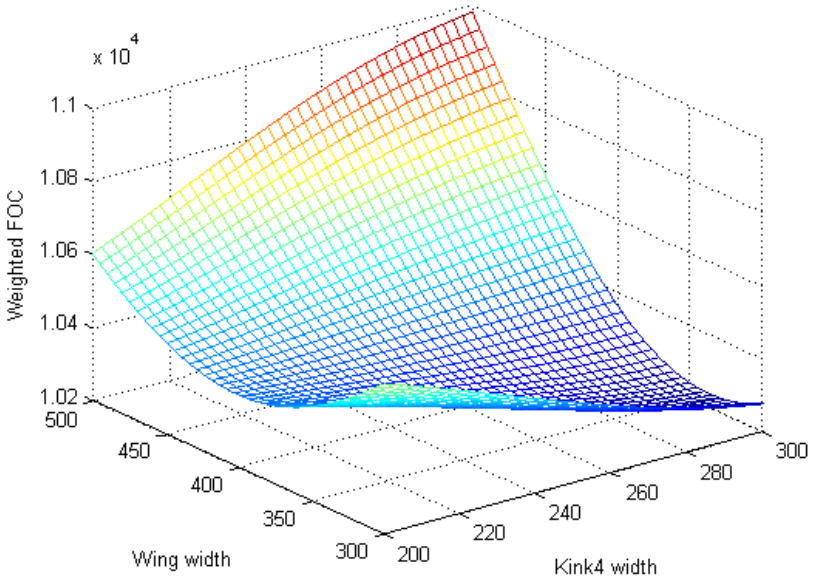
Surrogates of the weighted FOC over a slice of the design space are shown in Figure 5-7; specifically, the axes of this figure represent the widths of the two outermost sections, with the section chords and sweeps fixed. The root-mean square difference between the grid points used to construct the two models is approximately $356, representing a reduction in weighted FOC of about 3.5% for the optimal cruise altitude case. Considering that the system objective is the weighted FOC, or the weighted



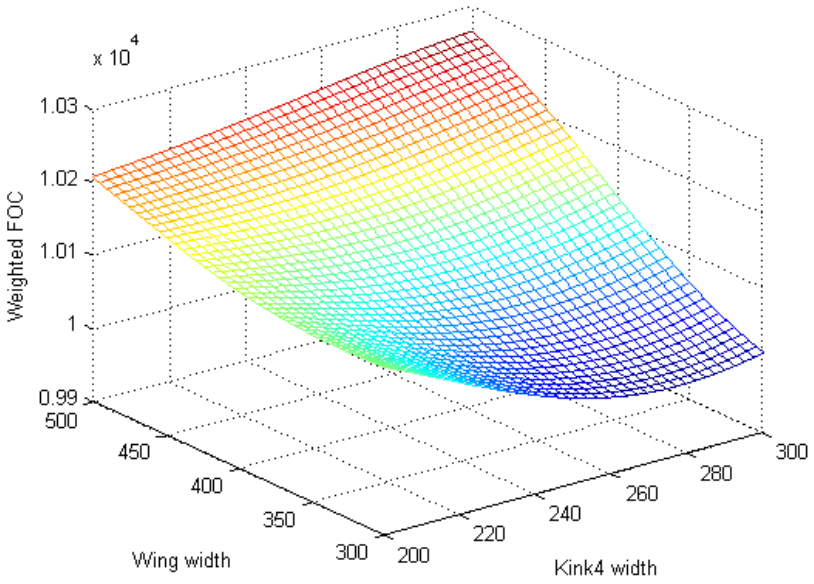
(a) (b)

Figure 5-6: Histograms of the optimal initial cruise altitude for (a) the short-range mission and (b) the medium-range mission calculated by the lower-level optimization.

cost to operate the aircraft for a mission, a 3.5% reduction in this cost can be substantial over the lifetime of the aircraft. Thus, application of the bi-level framework reduces the design space to one exhibiting much improved system metrics relative to a fixed cruise altitude study. Furthermore, the surrogate constructed after application of the bi-level framework shows considerably less variation than the one that fixed the initial cruise altitude to default values, indicating that the system objective after application of the bi-level framework is less sensitive to the planform geometry; this behavior is demonstrated in the previous two examples employing the framework, as well. Therefore, the bi-level framework offers the advantage of improving the design space exploration in terms of investigating configurations with better system performance, while often times decreasing the sensitivity of the objective to the upper-level design variables.



(*a*)



(*b*)

Figure 5-7: Surrogates of a slice of the design space for the weighted FOC using (a) default values for the optimal cruise altitude and (b) the bi-level framework to find the optimal cruise altitude for the two missions.

Chapter 6

Conclusions

6.1 Summary of Results

The blended wing body aircraft is a viable and potentially advantageous aircraft configuration which could meet the challenging environmental and performance requirements of the next-generation aircraft. Due to the BWB's highly integrated features, system performance is significantly affected by the couplings between the various disciplines. Significant gains can be made by modeling these interactions early in the design process. Thus, it is crucial that BWB design be performed using multidisciplinary methods. The objectives of this thesis centered around the development of multidisciplinary methods for performing trade studies on BWB configurations at the conceptual design phase of the design process.

To address these objectives, a multisiciplinary tool was developed which was intuitive and familiar to the designer and allowed rapid trade studies of BWB configurations at the conceptual design level. The BWB design tool combined low-fidelity analyses from the primary disciplines of aircraft design — aerodynamics, weights, stability and controls, and propulsion — to simulate the system and calculate quantities of interest such as fuel burn and cruise lift-to-drag ratio, as well as to estimate that the configuration is balanced. A design exercise demonstrated that the tool provides reasonable estimates of system performance metrics relative to previously developed methods used to conceptually design an existing BWB configuration. Moreover, two

example trade studies were performed using the BWB design tool which highlighted the potential utilization of the tool. Results of the example studies demonstrated that the BWB design tool captures the major effects of planform geometry changes on system performance and, additionally, highlighted the simplification of configuration modification that the tool allows.

In addition to the development of the BWB design tool, a framework was developed which provides a means to reduce the dimension of the design space while ensuring that good values are chosen for the parameters. By partitioning the original design variables into an upper- and lower-level set, and subsequently performing an optimization over the lower-level variables, this framework ensures that the lower-level variables are set to good values relative to some lower-level objective. The bi-level framework thus reduces the number of design variables over which the designer must explore while simultaneously ensuring that an effective trade study can be performed. The bi-level framework was demonstrated on several trade studies involving the exploration over BWB planform geometries. The first example problem considered the outerwing airfoil shape as a lower-level variable and optimized this shape for cruise drag; results showed that application of the bi-level framework generated a surrogate model for the fuel burn which represented an approximately 1% reduction in fuel burn relative to a default airfoil, while also allowing a full visualization of the design space. The other example problems used the BWB design tool to apply the bi-level framework; one problem considered the cruise Mach number as a lower-level variable, while the second problem used the mission trajectory as a lower-level design variable. Results of the two example problems demonstrated that the bi-level framework resulted in significant modifications to the shape of the design space, while providing a reduction in the system objectives of interest.

6.2 Recommendations for Future Work

A major discipline absent from the BWB design tool is a structural module. Such a module would provide better weight estimates and, in turn, more trustworthy es-

timates for the aircraft moments of inertia, thus giving higher fidelity calculations for the stability & controls estimates. Due to the uncoventional shape of BWB aircraft, low-fidelity methods to analyze the structural characteristics of a configuration are not well developed and most existing analyses rely on a structural finite element analysis (FEA) of the aircraft. However, the addition of a structural FEA within the BWB design tool would require significant improvements to the current geometric parameterization of the aircraft present in the tool, since an FEA requires specification of detailed variables describing spars and ribs, as well as the composites comprising the outer skin of the aircraft. Furthermore, a finite element analysis is at a higher level of fidelity than the majority of discipline analyses already integrated into the tool and would therefore have a significant effect on the computational intensity of a system simulation, while the quality of the simulation would still be driven by the lower-fidelity analyses. Also, the inherent coupling between the aerodynamics and structural module would introduce a feedback loop into the system analysis, further increasing the computational requirements of a full system simulation. However, this aero-structural coupling plays a major role in the performance of a BWB configuration and is thus vital for an exhaustive analysis of the aircraft.

Additional insight from the trade studies could be gained if high-fidelity models are added to the existing discipline analyses in the BWB design tool. Similar to the above discussion, the inclusion of a 3-D Navier-Stokes code, or a component-wise propulsion model, similar to that used in TASOPT, could significantly add to the accuracy in the flight profile estimation and thus the prediction of performance metrics. However, the inclusion of higher fidelity models into the BWB design tool could lead to a significant increase in the runtime of a single system simulation, undermining the original objectives of the tool. By utilizing recently developed multifidelity methods, the accuracy in the simulations could be increased without a large sacrifice in required computational time [26]. Therefore, it is certainly possible to add more accurate modeling capabilities to the BWB design tool without significantly sacrificing the original objectives of the tool.

Aside from enhancements to the models contained within the BWB tool, there

exist several potential avenues for future research concerning the bi-level framework. Specifically, the bi-level framework as developed in Chapter 4 of this thesis relies on the human designer to partition the design space of the system based on engineering experience and intuition. For many engineering systems, this paritioning can be made quite naturally. However, techniques do exist, mainly within the machine learning literature, that can estimate groupings amongst a set of variables, which could potentially find optimal partitionings for a design space; for example, the support vector clustering seems promising for such an exercise [5]. Thus, such methods could enhance the bi-level framework by automating the partitioning process (or at least informing the designer of potential partitionings) and further enriching the information gained during the trade space exploration.

Furthermore, the bi-level framework, as developed in this thesis, requires that the lower-level optimization problem be solved to optimality; that is, the x_ℓ^* computed for each upper-level sample be found by fully solving the lower-level optimization problem. In the case of non-convex lower-level objective functions or blackbox simulations which require extensive runtimes, finding a local optimum could prove quite challenging. Given that the goal of the bi-level framework is to find "good" values for the lower-level variables in order to perform effective trade studies on the system, it is certainly plausible that the bi-level framework could satisfy its intent with only partial, or approximate, optimizations of the lower-level variables. Specifically, if the chosen optimization routine guarantees a reduction in the objective function at each iteration, as is the case with many optimization procedures, then the bi-level framework could be applied without the computational burden of performing a full lower-level optimization, while still providing a beneficial effect on the exploration.

Bibliography

[1] Commercial Aircraft Directory - Aircraft Specification: Airbus A321-100. `http://www.flightglobal.com/directory/`. Accessed: March 2013.

[2] Jet Fuel Monthly Price - US Dollars per Gallon. `http://www.indexmundi.com/commodities/?commodity=jet-fuel`. Accessed: April 2013.

[3] J. Agte, O. de Weck, J. Sobieszczanski-Sobieski, P. Arendsen, A. Morris, and M. Spieck. MDO: assessment and direction for advancement – an opinion of one international group. *Structural and Multidisciplinary Optimization*, 40(1):17–33, 2010.

[4] P.P. Belobaba. Operating costs and productivity measures. 16.75J Class Lecture Notes, 2006.

[5] A. Ben-Hur, D. Horn, H.T. Siegelmann, and V. Vapnik. Support vector clustering. *The Journal of Machine Learning Research*, 2:125–137, 2002.

[6] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1):238–252, 1962.

[7] G.E.P. Box, J.S. Hunter, and W.G. Hunter. *Statistics for Experimenters: Design, Innovation, and Discovery*. Wiley Online Library, 2005.

[8] R.D. Braun. *Collaborative Optimization: An Architecture for Large-Scale Distributed Design*. PhD thesis, Stanford University, 1996.

[9] E.J. Cramer, J.E. Dennis, Jr, P.D. Frank, R.M. Lewis, and G.R. Shubin. Problem formulation for multidisciplinary optimization. *SIAM Journal on Optimization*, 4(4):754–776, 1994.

[10] S. Dempe. *Foundations of Bilevel Programming*. Springer, 2002.

[11] A. Diedrich. The multidisciplinary design and optimization of an unconventional, extremely quiet transport aircraft. Master's thesis, Massachusetts Institute of Technology, 2005.

[12] M. Drela. XFOIL: An analysis and design system for low Reynolds number airfoils. In *Low Reynolds Number Aerodynamics*, pages 1–12. Springer, 1989.

[13] M Drela. MSES Multi-Element airfoil design/analysis software: Summary. `http://raphael.mit.edu/drela/msessum.ps`, 2004.

[14] M. Drela. Simultaneous optimization of the airframe, powerplant, and operation of transport aircraft. In *RAeS 2nd Aircraft Structural Design Conference, Hamilton Place, London*, 26-28 Oct., 2010.

[15] M. Drela and H. Youngren. AVL 3.30 User Primer. `http://web.mit.edu/drela/Public/web/avl/avl_doc.txt`. Accessed: August 2011.

[16] O. Gur, W.H. Mason, and J.A. Schetz. Full-configuration drag estimation. *Journal of Aircraft*, 47(4):1356–1367, 2010.

[17] R.T. Haftka. Simultaneous analysis and design. *AIAA Journal*, 23(7):1099–1103, 1985.

[18] J.I. Hileman, Z.S. Spakovszky, M. F, M.A. Sargeant, and A. Jones. Airframe design for silent fuel-efficient aircraft. *Journal of Aircraft*, 47(3):956–969, 2010.

[19] W.F. Hilton, L. Bairstow, and G. Berk. *High-Speed Aerodynamics*. Longmans, Green and Company, 1952.

[20] A. Jones. Multidisciplinary optimization of aircraft design and takeoff operations for low noise. Master's thesis, Massachusetts Institute of Technology, 2006.

[21] A.B. Lambe and J.R.R.A. Martins. Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. *Structural and Multidisciplinary Optimization*, 46(2):273–284, 2012.

[22] R.H. Liebeck. Design of the blended wing body subsonic transport. *Journal of Aircraft*, 41(1):10–25, 2004.

[23] S.N. Lophaven, H.B. Nielsen, and J. Søndergaard. Aspects of the Matlab toolbox DACE. Technical Report IMM-REP-2002-13, Technical University of Denmark, August 2002.

[24] A.E. Magnus and M.A. Epton. PANAIR - A Computer Program for Predicting Subsonic or Supersonic Linear Potential Flows about Arbitrary Configurations Using a Higher Order Panel Method. Volume I. Theory Document (Version 1.0). Technical Report D180-24910-3-VOL-1, Boeing Military Airplane Co., Seattle, WA, 1980.

[25] A. March. Influence of low-speed aerodynamic performance on airport community noise. Master's thesis, Massachusetts Institute of Technology, 2008.

[26] A.I. March. *Multidelity methods for multidisciplinary system design*. PhD thesis, Massachusetts Institute of Technology, 2012.

[27] D. Marsh. Mass moment of inertia estimation methods – manned aircraft. In *The 21st National Conference of the Society of Aeronautical Weight Engineers. Seattle, WA*, 14–17 May, 1962.

[28] J.R.R.A. Martins and A.B. Lambe. Multidisciplinary design optimization: Survey of architectures. *AIAA Journal*, 2013. (In press).

[29] W.H. Mason. Friction. Technical report, Virginia Tech Aerodynamics and Design Software collection, February 2011. Available online at `http://www.aoe.vt.edu/~Mason_f/FRICTman.pdf`.

[30] Mathworks, Inc. Constrained nonlinear optimization. Optimization Toolbox User's Guide, V.5, 2010.

[31] Michael D McKay, Richard J Beckman, and William J Conover. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[32] F. Palacios, M.R. Colonno, A.C. Aranake, A. Campos, S.R. Copeland, T.D. Economon, A.K. Lonkar, T.W. Lukaczyk, T.W.R. Taylor, and J.J. Alonso. Stanford University Unstructured (SU^2): An open-source integrated computational environment for multi-physics simulation and design. In *51st AIAA Aerospace Sciences Meeting and Exhibit, Grapevine, TX, AIAA, AIAA 2013-0287*, 7-10 January, 2013.

[33] D.P. Raymer. *Aircraft Design: A Conceptual Approach*. Reston,VA: American Institute of Aeronautics and Astronautics, 4th edition, 2006.

[34] M.A. Sargeant, T.P. Hynes, W.R. Graham, J.I. Hileman, M. Drela, and Z.S. Spakovszky. Stability of hybrid-wing-body-type aircraft with centerbody leading-edge carving. *Journal of Aircraft*, 47(3):970–974, 2010.

[35] R.D. Schaufele. *The Elements of Aircraft Preliminary Design*. Aries Publications, 2000.

[36] R.S. Shevell. *Fundamentals of Flight*. Prentice Hall, Upper Saddle River, New Jersey, second edition, 1989.

[37] T.W. Simpson, D.K.J. Lin, and W. Chen. Sampling strategies for computer experiments: design and analysis. *International Journal of Reliability and Applications*, 2(3):209–240, 2001.

[38] T.W. Simpson and J.R.R.A. Martins. Multidisciplinary design optimization for complex engineered systems: Report from a national science foundation workshop. *Transactions of the ASME-R-Journal of Mechanical Design*, 133(10):101002, 2011.

[39] J. Sobieszczanski-Sobieski. Optimization by decomposition: A step from hierarchic to non-hierarchic systems. Technical Report NASA-TM-101494, NASA, September 1988.

[40] J. Sobieszczanski-Sobieski, J.S. Agte, and R.R. Sandusky. Bilevel integrated system synthesis. *AIAA Journal*, 38(1):164–172, January 2000.

[41] J. Sobieszczanski-Sobieski, T.D. Altus, M. Phillips, and R. Sandusky. Bilevel integrated system synthesis for concurrent and distributed processing. *AIAA Journal*, 41(10):1996–2003, October 2003.

[42] D.V. Steward. Partitioning and tearing systems of equations. *Journal of the Society for Industrial & Applied Mathematics, Series B: Numerical Analysis*, 2(2):345–365, 1965.

[43] G. Stump, S. Lego, M. Yukish, T.W. Simpson, and J.A. Donndelinger. Visual steering commands for trade space exploration: User-guided sampling with example. *Journal of Computing and Information Science in Engineering*, 9(4):044501, 2009.

[44] S. Tosserams, L.F.P. Etman, and J.E. Rooda. A classification of methods for distributed system optimization based on formulation structure. *Structural and Multidisciplinary Optimization*, 39(5):503–517, 2009.

[45] W.O. Valarezo and V.D. Chin. Method for the prediction of wing maximum lift. *Journal of Aircraft*, 31(1):103–109, 1994.

[46] S. Wakayama. Blended-wing-body optimization problem setup. *AIAA Paper*, 2000-4740, September 2000.

[47] E.H. Winer and C.L. Bloebaum. Visual design steering for optimization solution improvement. *Structural and Multidisciplinary Optimization*, 22(3):219–229, 2001.

[48] E.H. Winer and C.L. Bloebaum. Development of visual design steering as an aid in large-scale multidisciplinary design optimization. part i: method development. *Structural and Multidisciplinary Optimization*, 23(6):412–424, 2002.