BAYESIAN COLLABORATIVE SAMPLING: ADAPTIVE LEARNING FOR MULTIDISCIPLINARY DESIGN

A Thesis
Presented to
The Academic Faculty

by

Chung Hyun Lee

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

Georgia Institute of Technology
December 2011

BAYESIAN COLLABORATIVE SAMPLING: ADAPTIVE LEARNING FOR MULTIDISCIPLINARY DESIGN

Approved by:

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

Professor Brian German
School of Aerospace Engineering
Georgia Institute of Technology

Professor Jechiel Jagoda
School of Aerospace Engineering
Georgia Institute of Technology

Dr. May-fun Liou
Multidisciplinary Design, Analysis, and Optimization Branch
NASA Glenn Research Center

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

Date Approved: 11 November 2011

ACKNOWLEDGEMENTS

I would first like to thank my advisor, Dr. Mavris, for the years of guidance and support. I came to Georgia Tech with a rather narrow notion of engineering research, and Doc has widened my view of what is needed to be both a successful engineer and researcher. I would also like to thank the members of my committee: Dr. German, Dr. Jagoda, Dr. Liou, and Dr. Tai. Their advice and difficult questions have helped this dissertation – but more importantly, they have taught me to become a more careful and thoughtful student of engineering.

Thanks to my friends at ASDL, especially Jonny Murphy and Miguel Walter, who were exceedingly helpful and patient as we discussed many of my half-baked ideas. I am indebted to Kemp Kernstine for pointing out some theoretical issues relating to the sparse Bayesian models.

This research was supported by the NASA Graduate Student Research Program. The Multidisciplinary Analysis and Optimization (MDAO) Branch at NASA Glenn Research Center is full of warm people and was an exciting place to do research for thirty weeks. Thanks to Dr. May-fun Liou, Dr. Meng-sing Liou, Mr. Justin Gray, Mr. Scott Townsend, and the rest of the branch.

Finally, I want to thank my family for their love and support. Also, my brother actually sat down with pen and paper and worked through most of the mathematics in this thesis. Any remaining errors unfortunately cannot be blamed on him.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

A Bayesian adaptive sampling method is developed for highly coupled multidisciplinary design problems. The method addresses a major challenge in aerospace design: exploration of a design space with computationally expensive analysis tools such as computational fluid dynamics (CFD) or finite element analysis. With a limited analysis budget, it is often impossible to optimize directly or to explore the design space with design of experiments (DoE) and surrogate models.

In such cases, the designer can conserve function calls by eliminating 'bad' regions within some confidence level. Exploration can then be concentrated in regions that probably contain a global optimum, in light of current data. However, current sampling methods, such as design of experiments (DoE), can be inefficient by attempting to represent all regions uniformly. Offline sampling techniques like DoE require selection of an entire input sample before evaluating any of them. If the analyses are executed serially, these strategies ignore valuable information since remaining sample points cannot be chosen in light of accumulating knowledge. As a result, such sampling methods can continue to run analyses in a 'bad' region long after there is sufficient certainty about its 'badness.' This inefficiency is magnified in multidisciplinary problems with feedbacks between disciplines since each design point may require iterative analyses to converge on a compatible solution between different disciplines.

To address these problems, this thesis describes Bayesian Collaborative Sampling (BCS). BCS is a bi-level architecture for design space sampling that uses Bayesian models with online, or adaptive, sampling. Bayesian adaptive sampling methods already exist, but BCS is tailored for strongly coupled multidisciplinary problems. It

is novel because it simultaneously does the following:

1. concentrates disciplinary analyses in regions of a design space that are favorable to a *system*-level objective

2. guides analyses to regions where interdisciplinary coupling variables are probably compatible

BCS uses Bayesian sequential learning techniques along with elements of the collaborative optimization (CO) architecture for multidisciplinary optimization (MDO). Like CO, BCS decomposes a multidisciplinary problem into a system level and discipline level. In addition, it equips both of these levels with Bayesian models. In particular, BCS can borrow recent advances from machine learning literature and can use sparse Bayesian models for computational speed. At each iteration, BCS uses sequential learning criteria called *expected improvement* and *probability of target match* from the Bayesian models to select the next sample point for analysis.

The method is first tested with a subsonic glider wing design problem using low fidelity analysis. The results show favorable performance compared to an off-line DoE. Further performance tests investigate the method's dependence on coupling bandwidth, warm-start sample size, and replication error. Finally, BCS is demonstrated with CFD in coupled aero-propulsion design of a turbojet engine nacelle. Successful tests show that the BCS can be used for practical design with high fidelity analysis codes.

CHAPTER I

INTRODUCTION

Engineering design often involves tradeoffs between a multitude of disciplines, such as aerodynamics, structures, and propulsion analysis in aircraft design. Due to interactions between the physics of each discipline, the best design for overall system performance may be different from the best within each discipline. The goal of multidisciplinary design is to use discipline-level analyses efficiently and exploit their interactions to find good system-level performance.

This thesis is motivated by two common challenges that are encountered in many multidisciplinary design problems:

- the need for computationally expensive, high fidelity physics modeling in early design
- the large number of analyses required to converge disciplines on physically compatible solutions for each individual design (e.g. aerodynamic lift equals structural weight)

These problems can be illustrated in terms of a modern design example. Recently, the boundary layer ingesting (BLI) hybrid wing-body (HWB) aircraft concept has received both academic and industry attention [12],[107],[115],[122],[126]. One baseline design from NASA is shown in Figure 1 [50]. The BLI HWB has flush mounted engine nacelles that ingest boundary layer air. This configuration may achieve aerodynamic advantages due to engines re-energizing the boundary layer on the aircraft upper surface and filling the aircraft wake with exhaust plumes. The tight coupling between aerodynamics and propulsion physics may be exploited to achieve lower fuel consumption.

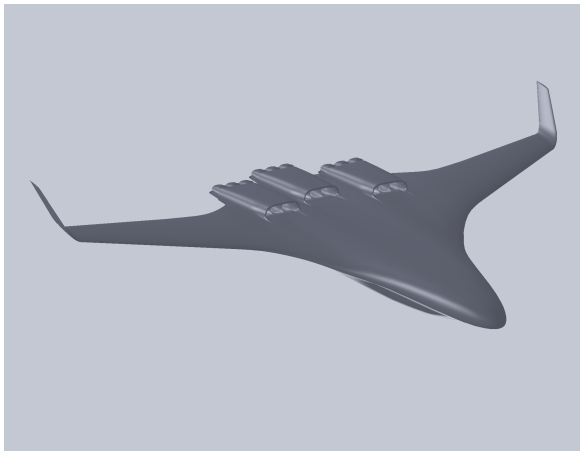

Figure 1: Boeing/NASA N2B boundary layer ingesting hybrid wing-body

First, note that the main advantages of the BLI concept inherently rely on complex compressible and viscous flow physics that are best captured by high fidelity models. Traditional design processes often begin with combination of historical data and low fidelity models such as potential flow for early design and then use higher fidelity models in later, detailed design. However, to fully exploit the advantages of boundary layer ingestion and the wake filling concept, one must have sufficient fidelity in physics models even in early stages of design since they influence major design commitments such as wing area or number of engines. If early aircraft-propulsion sizing is done with lower order physics models, then the boundary layer and wake advantages can only be exploited later by minor, detailed design adjustments from a suboptimal baseline.

Unfortunately, in 2011, accurate aerodynamic analysis of a BLI aircraft with tools like Reynolds-Averaged Navier Stokes (RANS) CFD has a very high computational cost. Each complete CFD function call can require thousands of CPU hours on a high-performance supercomputing cluster. There is therefore a need for methods to bring information from computationally intensive physics models to early design in cases where the analysis budget is severely limited.

In addition, the BLI HWB aircraft design example shows a second theoretical challenge related to the coupled physics between disciplines. The main intent of the

BLI concept is to exploit a coupling between the external airframe aerodynamics and the propulsion cycle analysis. The BLI engines ingest "bad" boundary layer air with significant total pressure loss. The low total pressure at the inlet causes the turbofans to operate relatively inefficiently and produce less thrust than otherwise. However, this inefficiency and low thrust is compensated by improved airframe aerodynamics and lower power dissipation in the plume/wake. (A more thorough description of the physics will be given in a later chapter.)

Because of this strong aero-propulsion coupling, early design for BLI aircraft cannot be done with independent aerodynamics and propulsion analyses as is often done in traditional aircraft design. In a conventional, tube-and-wing aircraft configuration, the engines are placed under and forward of the wings to minimize the mutual interference between the airframe and engines. The unpowered airframe aerodynamics can often be analyzed separately to find drag (thrust required). This required thrust is passed to the propulsion discipline, which may try to optimize engine efficiency while meeting the required thrust. Mutual influence of the two disciplines may reasonably be ignored in early stages; for example, the plume has little effect on the airframe aerodynamics, and airframe aerodynamics has little effect on the propulsion cycle. However, in a BLI aircraft, the airframe lift and drag depend heavily on the operation of the engine, since the engine is immersed in the boundary layer/wake. At the same time, the engine depends heavily on the airframe boundary layer and wake since it ingests sluggish air that has lost energy and total pressure from its free-stream condition.

In practice, the strong mutual influence between disciplines must be enforced by iteratively guessing values for coupling variables that link the two until they become physically consistent. For example, the aerodynamics discipline must guess CFD boundary conditions that produce mass flow, inlet and nozzle conditions that the propulsion discipline expects. The propulsion discipline must use this mass flow and

guess propulsion cycle parameters to match the external aerodynamics boundary conditions. Iterative solutions are required to achieve *interdisciplinary compatibility* – the state when coupling variables are consistent between disciplines. This iteration may require a large number of computationally expensive analyses to find the performance of a single design.

It may be reasonable to simply assume compatibility with tube-and-wing aircraft when this coupling is weak. However, the mutual influence is the main source of benefit for the BLI HWB, so interdisciplinary compatibility must be explicitly modeled and enforced as early as possible in the design process.

1.0.1 Chapter Roadmap

To summarize again, this thesis is motivated by design problems like the boundary layer ingesting hybrid wing-body where there are:

- a need for computationally intensive, high fidelity physics models in early design
- a need to enforce interdisciplinary compatibility when mutual influence between disciplines is too significant to be ignored

In the rest of this chapter, each motivation and the shortcomings of current methods to address it are presented and discussed in further detail. At the end of the chapter, a general description is given for the thesis's main method to address them. Bayesian Collaborative Sampling (BCS) is an adaptive sampling method that guides disciplinary analyses toward favorable system objectives as well as compatibility between disciplines. A more detailed, formal description of BCS will be given after providing theoretical background in Chapter 2.

1.1 Motivation 1: Harnessing Computationally Expensive Analysis Codes

A major challenge in multidisciplinary design is the efficient use of computationally expensive analysis tools in early design phases. Despite major advances in computer speed and simulation, high fidelity analysis codes like computational fluid dynamics (CFD) and finite element analysis (FEA) are often too costly for many large-scale applications. The expense can include not only computer processor time, but also the time required to set up parametric environments for automated execution and post-processing, queuing time on a multi-user high performance computing (HPC) environment, and even the actual monetary expense of multiple software licenses. Regardless of computational advances, there will almost always exist an "expensive" analysis tool. Driven by competition, engineers will seek more accurate or thorough analysis in less time than rivals, as long as additional information can be exploited for performance advantage.

Currently, computationally intensive tools like CFD are not often used for design space exploration or optimization in conceptual or preliminary design stages. Instead, they are usually reserved for detailed design — after the major design decisions have already been settled and large changes are costly. Early, important decisions must instead rely on simplified physics, reduced order models or historical data. However, there are many contemporary engineering problems that are dominated by complex physics that can only be modeled by computationally expensive codes. Second, revolutionary designs may not be able to call on historical data. It is therefore crucial to develop ways to use higher fidelity tools efficiently in those early design phases where high fidelity information can have a larger impact on the final design.

So, with novel concepts like a BLI aircraft, designers may want very expensive, high-fidelity simulation tools in early design. Other design examples may include aero-acoustic design of unducted rotors, hypersonic air-breathing propulsion, and

ultra-low Reynolds number vehicles. To tackle such complex problems, aerospace design research in recent decades has pursued two main methodological paths: 1) design space exploration with surrogate modeling and 2) optimization methods.

First, surrogate modeling is the fitting of a simplified equation to a representative sample of inputs and their analysis outputs (responses) — essentially curve-fitting to approximate the analysis functions. Surrogate models include simple polynomial response surfaces or more complicated artificial neural networks (ANN) or support vector machines (SVM). This method is useful because it provides a global view of the design space and is often more informative than pure optimization. For example, if the design space contains no feasible region, optimization would fail. However, a surrogate model would allow the designer to study the effects of relaxing constraints on the feasible region or to investigate the effect of technology infusion by including performance scaling factors in the surrogate.

Second, design research has also focused on optimization methods. This category encompasses basic gradient-based optimizers to heuristic/stochastic algorithms such as genetic algorithms and simulated annealing. The category includes multi-disciplinary optimization methods that arrange information flow between engineering disciplines to efficiently minimize an overall objective; it also includes multi-objective optimization methods that find a Pareto front (a surface of non-dominated solutions that optimize two or more goals).

Both surrogate modeling and optimization can be useful design tools. However, both can encounter difficulties when the analysis codes are computationally expensive. Surrogate models require dense samples to represent an underlying function with global accuracy. This requirement is magnified by the "curse of dimensionality," in which the volume of a design space increases exponentially with the number of design variables. Adding a single design variable can lead to a very large increase in the sample size required for the same model accuracy. Direct optimization is also

affected by this curse of dimensionality and can require a large number of function calls to converge. In addition, it is difficult to make a prior estimate of the number of function calls required for an optimization. If the analysis budget is depleted before finding an optimum, the intermediate result may not be very useful; for example, the optimizer may not even find a feasible region. Designers can often work around this problem by relying on low or medium-fidelity tools to narrow the ranges of the design space. However, this option is not always available, for the main physical phenomena may be irreducibly complex. For example, in a transonic compressor blade design, shock-induced flow separation is one of the main physical phenomena that affects performance. So, the design cannot proceed far without high fidelity analysis of compressible and viscous fluid mechanics.

In addition to these practical problems of computational cost, there is a perhaps a more philosophical question of how to use high fidelity analyses in early design. How useful is an optimum solution or a globally accurate surrogate model in the conceptual/preliminary design? In these early phases, there is a fundamental lack of knowledge about the design that limits the certainty of analysis or optimization, regardless of fidelity level. This lack of knowledge includes both *known unknowns* and *unknown unknowns*. Known unknowns include minor variables that are screened out from design studies or physical assumptions made during analysis. In a wing design, for example, wing root fillets and rivet locations might be ignored initially, and CFD analyses might be conducted only at one of several possible flight conditions. Such known unknowns may be addressed only later in detailed design, so there exists uncertainty about final results after the rivets and fillets are added. In the second category — unknown unknowns — some variables can only be identified after design knowledge accumulates. For example, preliminary design studies may reveal that there is no feasible aircraft design that can meet a takeoff field length (TOFL) constraint. High lift devices like leading edge slats or boundary layer suction may then

be introduced. The variables describing the high lift device are unknown unknowns in early phases, because the designer does not even know that they will be part of the final wing design.

Given both types of unknowns in early design phases, the results of surrogate modeling and optimization are highly uncertain in early stages even with perfect fidelity. Neither can be used as a push-button, automatic tool to select a final design for production. Rather, the design tools are useful because they reveal important trends, interactions, and sensitivities. For example, if a wing optimizer converges on a wing with 15% thickness to chord ratio, that ratio is not certainly the best (although the final production design will probably have a similar value). However, it is interesting that the "optimum" wing is so thick. By examining the optimization process, the designer may discover that a mission range requirement led the optimizer to select a thick wing for fuel storage volume rather than for aerodynamic reasons. This sort of qualitative information is often more important than precise values. It shows the designer that the *concept* of a thick wing is optimal, even though the 15% value may be uncertain.

The first main motivation of this research project is, once again, the need to bring high-fidelity analysis to conceptual and preliminary design. With such computationally expensive tools, it is often impossible to create an accurate surrogate model or converge on an optimum. However, finding a globally accurate model or a precise optimum are not the designer's main goal—the value of such tools are limited anyway because of the fundamental lack of knowledge in early design. So, instead of striving to fit accurate models or find an optimum with fewer function calls, a method is needed to glean the most information from a very limited budget of function calls.

The Bayesian collaborative sampling (BCS) method addresses this need with a hybrid of surrogate modeling and MDO. It involves adaptively sampling a multidisciplinary design space, balancing between the exploration of unknown regions and

exploitation of favorable trends while seeking a dense sample near the global optimum. "Bad" regions are sampled just sufficiently to establish that they are probably bad. "Good" regions are sampled densely until it is discovered that global optimum probably lies elsewhere. This goal-driven design space exploration progressively eliminates regions of the design and forms an increasingly accurate estimate of the feasible, favorable region. BCS may not converge on a single optimum design; nor will it produce a globally accurate surrogate. However, it will show which regions are most *likely* to contain improvements and which regions are *probably* infeasible or unfavorable. While the final result is not as satisfying as an accurate surrogate model or a converged optimization, such probabilistic methods may be a more practical when other methods are unaffordable. The designer attempts to gain the maximum information about the global optimum at any given number of iterations. For example, if the algorithm is arbitrarily stopped after N function calls, the proposed strategy aims to give the best probabilistic view of the global optimum that can be purchased with N function calls.

The current state and limitations of surrogate modeling and MDO are described before discussing how the present thesis builds on them.

1.1.1 Surrogate Modeling (Metamodeling)

Surrogate modeling is a common approximation technique in aerospace design. A representative sample of points is selected from a space of design variables, and each point is evaluated with analysis codes. The resulting sample is then used to fit linear or non-linear multivariate regression or interpolation equations [103]. These analytic expressions are surrogate models (or 'meta-models'), and they can usually be evaluated in a fraction of a second. Examples of surrogate models include polynomial response surfaces, neural networks, Kriging/Gaussian processes, radial basis functions, and support vector machines [81], [139], [142]. Surrogate models provide rapid

approximations of expensive analysis codes for optimization or other design tasks. The technique can be more useful than pure optimization alone, because it provides a global view of the design space. With a surrogate model, the designer can see the relative importance of design variables, see the effect of relaxing constraints, and understand general trends.

The quality of the training sample affects the accuracy of the surrogate model. Design of experiments (DoE)—or a systematic sampling plan—is often used to generate a data set to reveal the most information about the unknown, underlying function with the fewest function evaluations. In recent engineering design literature, it is very common to use DoE in conjunction with surrogate modeling. For current examples, see [13], [58], [79], [96].

1.1.2 Limitations of DoE and Surrogate Modeling for Computationally Expensive Applications

The simple, two-step DoE and surrogate modeling method is useful, but it has limitations when attempting MDO with expensive codes.

First, DoE/surrogate modeling is "cheap," but still not cheap enough — although it is a relatively efficient method to obtain a global view of a design space with fewer function calls, it still requires a substantial number of function calls. The method works well if there is a sufficient sample size to "train" or regress surrogate models to acceptable accuracy. However, this is often not possible with tools like CFD. In the author's experience, for a modest aerospace design problem of about $O(10)$ design variables, a sample size of $O(10^4)$ can be insufficient to regress surrogate models with acceptable model representation errors (standard deviation of less than 5% of response range). This is particularly true of highly nonlinear responses common in transonic aerodynamic applications. If the analysis is as expensive as large eddy simulation (LES) or even a wind tunnel experiment, accurate surrogate modeling may be impossible.

Second, DoE/surrogate modeling can be inefficient with function calls by being unnecessarily accurate in unpromising regions of the design space. A typical DoE is an 'offline' sampling method—in other words, it selects the entire set of sample input points before any of the points are evaluated. After the points are evaluated, a surrogate model is then fit to the data with the goal of minimizing error in the entire domain. Usually, an unfavorable region of the design space may have the same sampling density as favorable regions. This can be wasteful since only a small number of sample points are actually required to show with sufficient certainty that an unfavorable region does not contain an optimum. Note that the designer is not always concerned about only favorable regions. If there is a sufficient budget of analysis function calls, it is often desirable to run an DoE to evenly sample both favorable and unfavorable regions. This allows the designer to study global trends or to observe the impact of imposing constraints. However, with a limited budget, there is often little hope of achieving a globally accurate representation of the design space, and it is often useful to focus on a favorable region.

Figure 2 below shows caricature diagrams of: 1) the DoE/surrogate modeling approach and 2) an adaptive sampling method that clusters points for accuracy near the optimum. The dots are the sample points evaluated with the underlying true function, y. The black curve is a surrogate model, and the shaded region represents some measure of model uncertainty. For a similar number of points, the example

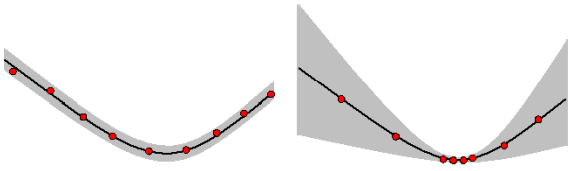

Figure 2: DoE/surrogate modeling (left) and adaptive sampling (right)

on the right uses analysis function calls more efficiently to represent the function's minimum. Even though there are regions of large uncertainty in the model, one

may be sufficiently certain that they do not contain an optimum. So, more analyses can be concentrated in the favorable region. Unfortunately, such a strategy is not possible with an offline sampling method like DoE, since there is no way to discern the favorable region prior to executing any analyses.

Unlike a DoE, an adaptive sampling method (also called "online sampling," "infill sampling," "sequential learning," or "query learning") selects and evaluates each new sample point based on observed data. The N-th input point is selected by exploiting information about the location and response values of all $N-1$ previous points.

Researchers have studied several types of adaptive sampling, usually in support of optimization rather than just rich sampling of a favorable region.[1] In a recent survey, Shan and Wang classify these as "metamodel-based design optimization" (MBDO) strategies [139]. First, a common practice is to alternate between modeling and optimization. The designer starts with a small sample DoE and surrogate model with wide ranges on design variables. The optimum of the model is found, the ranges are reduced around the model optimum, and another DoE and surrogate model are created. In one recent example of this method, Wang et al. describe the *adaptive response surface method* (ARSM), which progressively reduces the design variable ranges based on whether model response exceeds some threshold value [156]. Another iterative sampling and optimization method is to 1) fit a surrogate model, 2) optimize the surrogate model, 3) evaluate the analysis function at that model optimum, and repeat until model optimum and its actual response converge. For example, Vavalle and Qin recently used such a strategy in "response surface-based optimization" (RSO) with polynomial RSEs [155].

These methods have limitations. The chief problem is that they do not explicitly use model uncertainty to guide adaptive learning and may miss the global optimum

[1] Depending on one's point of view, these may be considered optimizer-based sampling or sampling/model-based optimization.

in a deceptive problem. Consider the caricature example in Figure 3 with a single function f of a single variable, x. Starting from the top left, suppose an adaptive

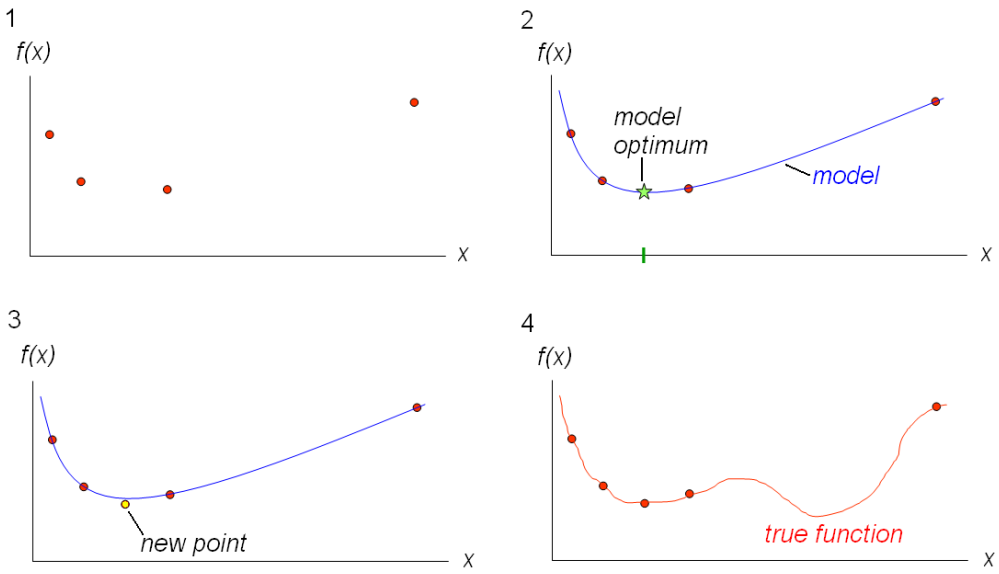

Figure 3: Adaptive sampling with surrogate model

learning/optimization algorithm begins with four initial designs of x evaluated for the true function $f(x)$. At top right, a surrogate model is fit to these points, and an optimizer finds the model optimum. This model optimum seems promising for the true optimum, so at bottom left, that point is evaluated with the true function (unfilled circle). This true response is slightly different from the model optimum, so the model is updated, and the adaptive learning process continues. However, at bottom right, it is apparent that the adaptive learning process has clustered sample points near a local optimum, but has missed the global optimum.

In this simple example, the problem is that there is a large, unsampled region where the model prediction is highly uncertain. The example adaptive learning algorithm does not account for the greater possibility that the model is incorrect and that the region contains a global optimum. It is easily fooled by this deceptive function. An improved adaptive learning algorithm would balance between exploitation and exploration. It would exploit regions that are densely sampled and show a favorable

trend — and also explore regions that may hide an optimum because they are not well known.

A second type of adaptive sampling strategy involves move limits or trust regions. A surrogate model is used for optimization until the optimizer moves outside of a region where the model is considered accurate. Then, more sample points are evaluated and the model is updated to improve accuracy where it is needed by the optimizer. For example, Dennis and Torczon discuss a trust region method for pattern search [39]. Alexandrov *et al* also describe such approximate model management techniques to coordinate the use of low and high fidelity analysis in optimization [6]. Other studies include Serafini [138] and Rodriguez *et al* [125]. Trust region methods accounts for local accuracy of surrogate models, but does not directly balance between exploitation and exploration of new regions. In theory, this might be accomplished by combining trust region methods with some global metaheuristic optimizer like a genetic algorithm, but the exploration would be stochastic rather than deliberately aiming for the most informative sample point.

A more recent, third adaptive sampling strategy is to use Bayesian modeling and inference to guide sampling toward an optimum based on some probabilistic criterion. A Bayesian model is a class of surrogate models that gives a predictive distribution on the response of a function. Using such model and its predictive distribution, the "probably best" next sample point can be inferred from currently observed points. A Bayesian adaptive algorithm accounts for local uncertainty. Mockus described an early Bayesian approach that uses conditional probabilities as a sampling criterion for new points [102], and this method was extended by Cox and John [33]. Jones, Schonlau and Welch described a similar method called Efficient Global Optimization (EGO) that uses Kriging (one type of Bayesian model) to calculate an expectation of improvement as a criterion to guide sampling toward an optimum [77], [131]. Variants of this Kriging-based method have been discussed by researchers at the University of

Southampton [52], [53], and others [152], [130].

These Kriging-based Bayesian methods balance between exploration of sparsely sampled regions and exploitation of accurately sampled regions, and are less likely than simple adaptive surrogate models to fail in deceptive problems as in Figure 3. While such methods are attractive for global optimization, they tend to be very computationally expensive. This is their principle drawback, and most of these studies have been limited to single blackbox functions rather than multidisciplinary problems.

The present thesis builds on these Kriging-based sampling strategies, adapting them for computationally cheaper Bayesian models and multidisciplinary problems. Because Kriging methods are an important pre-cursor to BCS, a more detailed literature review is given in the next chapter. The large computational expense of Kriging is addressed in this thesis by faster, sparse Bayesian regression techniques described in Chapter 3.

1.2 Motivation 2: Reducing the Cost of Interdisciplinary Compatibility

The previous discussion of sampling of expensive analyses applies generally to design problems where costly physics modeling is required. Multidisciplinary design analysis introduces another source of analysis cost in the form of *interdisciplinary compatibility*. This poses a special challenge for sampling of expensive codes.

1.2.1 Compatibility and Fixed Point Iteration

This compatibility is the state in which coupling variables passed between disciplines in a multidisciplinary analysis are converged and physically consistent. For example, consider an aero-structural analysis of a wing. A given design specifies the external geometry or outer mold line (OML) as well as the internal structure of spars, skin, and ribs. CFD analysis is performed on the nominal wing shape at some initial guess for angle of attack, α. The computed aerodynamic loads are passed to the structures

discipline. Finite element analysis is performed to compute structural deformations, and structural weight is estimated from material properties. The deformations are likely to be inconsistent with the initial shape used in CFD, and the weight may not balance with the aerodynamic lift. The design analysis suffers from *interdisciplinary incompatibility*.

This general type of problem is called fixed point iteration (FPI), which includes problems of the form $x = f(x)$. In this case, for example, the shape guessed by the aerodynamics discipline must equal the shape that results in the structures discipline due to computed aerodynamic loads. This general situation is represented by the circular path shown in the design structure matrix (DSM) in Figure 4.

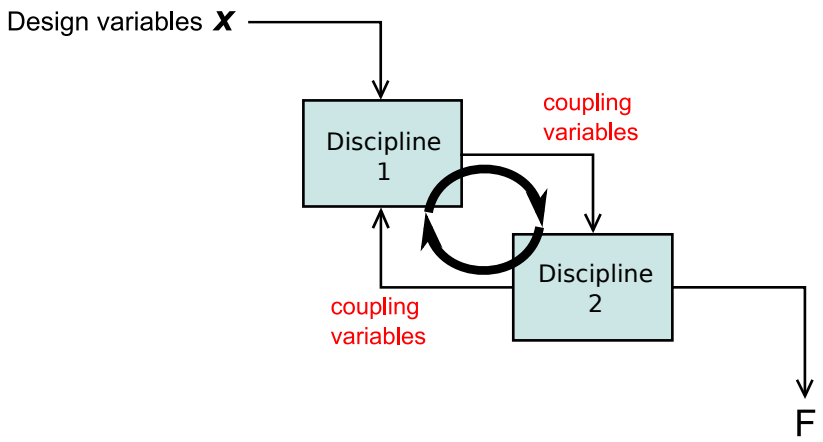

Figure 4: Design structure matrix with fixed point iteration (FPI)

The fixed point iteration problem can be solved by various solution methods. The simplest is to progressively update the inputs of each discipline from the output of others, and repeating the process until convergence is achieved. Other well-known methods include the root-finding methods of Newton or Halley. Note that the FPI problem must be solved iteratively to find the performance of each candidate design. In the wing analysis example, CFD must be performed many times on the same design at different α and deformed shapes just to find the lift/drag ratio of that single wing.

It is this repeated analysis for FPI that makes multidisciplinary design particularly expensive. A common engineering solution is to simply ignore the FPI problem by

cutting the feedback loop in the DSM. The feed-forward DSM can then be treated as a single black-box function, where each total design analysis requires only a single execution of each discipline.

This strategy is reasonable if the two-way coupling between disciplines is negligible – for example, if the wing is short and relatively rigid, the effect of structural deformations on the CFD aerodynamic loads can be small enough to be ignored. However, ignoring the coupling and FPI problem can undermine the basic intent of a design if the concept inherently exploits coupling of physics between disciplines.

This compatibility or fixed point iteration problem concerns each multidisciplinary *analysis* for an individual design. But in recent decades, methods have been developed to alleviate the problem in the context of *optimization*. The task of optimization rather than mere multidisciplinary analysis allows efficient techniques. For example, compatibility may not be enforced strictly at the beginning of the optimization, but may be enforced later as the optimum is approached. Such methods in multidisciplinary optimization (MDO) mainly focus on rearranging the flow of information between disciplines to efficiently enforce compatibility and optimization in concert.

BCS combines Bayesian methods with an MDO technique to address the interdisciplinary compatibility problem while adaptively sampling toward a favorable objective. BCS cannot be directly compared to MDO architectures, because it provides more information about the design space than a simple optimum. It also need not converge on an optimum to be useful because it seeks to richly sample the regions near the optimum. However, BCS exploits many of the same advantages as MDO architectures and actually uses a particular type — collaborative optimization — to guide its adaptive sampling. So, the current state-of-the-art and limitations of MDO architectures are discussed here.

1.2.2 MDO Architectures - Status and Limitations

MDO architectures are reformulations of an optimization problem for efficient information flow between analysis disciplines and numerical optimizers. In a typical engineering firm, the activities are usually split into disciplines of expertise, such as an aerodynamics group and a structures group. In smaller scale design, an analogous split may be made into discipline-related analysis codes. The purpose of an MDO architecture is: 1) to coordinate these separate disciplines in a search for a system-level optimum design, 2) to eliminate unnecessary information exchange, and 3) to minimize discipline function calls. Example architectures in recent literature include optimizer-based design (OBD), concurrent subspace optimization (CSSO), bi-level integrated system synthesis (BLISS), analytic target cascade (ATC), and collaborative optimization (CO)—see references [146], [121], [147], [100], [22]. Figure 5 shows notional design structure matrix (DSM) diagrams of OBD and CO architectures for wing design. The details are unimportant; these snapshots simply illustrate that MDO architectures provide a structure of information flow rather than the specifics of numerical algorithms.

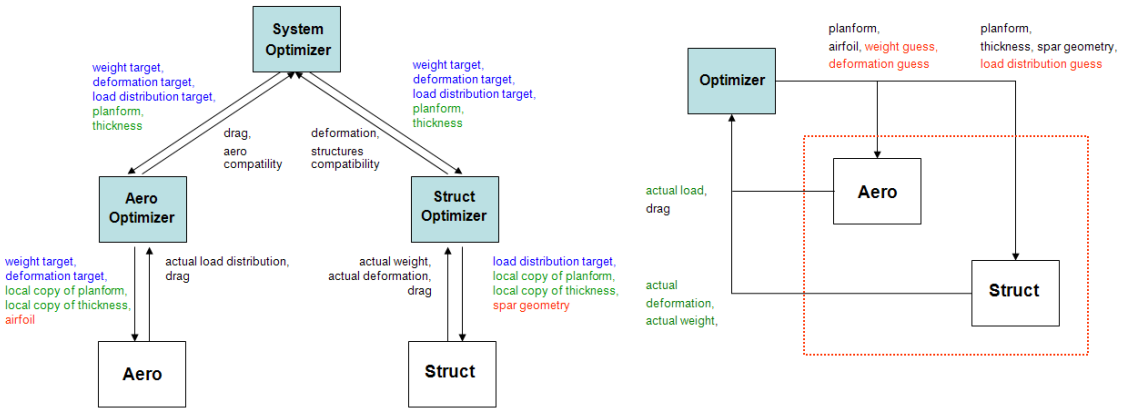

Figure 5: Two example architectures for the same aerostructural wing design problem: Collaborative Optimization (CO), left, and Optimizer Based Design (OBD), right

Simple architectures like OBD—also called "all-at-once" (AAO) optimization—give the control of all design variables to a single, central optimizer. In these architectures, the disciplines merely execute analysis for candidate designs handed down to them by the optimizer. Such approaches can be wasteful because they take away disciplinary autonomy and therefore diminish the value of skill and expertise in the disciplines. To address this problem, more recent architectures like collaborative optimization (CO), bi-level system synthesis (BLISS) and concurrent subspace optimization (CSSO) decompose the MDO problem into a system level and a discipline level. The system level is concerned with pursuing a main objective function (e.g. mission range, loiter time, return on investment), and it coordinates disciplines (e.g. aerodynamics, propulsion, structures) to subsidiary goals that support this objective. This decomposition allows groups in a typical engineering team to work within their existing organizational structure and also gives them autonomy to exploit their skill and best practices.

To save analysis cost, these multi-level architectures have recently been modified to use surrogate models. Some variants also incorporate adaptive learning; in each iteration, the results of discipline analyses are used to update disciplinary surrogate models. As the MDO process discovers potentially optimum regions, more disciplinary analyses are directed to the vicinity, and models are adapted. This can lead to higher accuracy of surrogate models in favorable regions of the design space. It also lowers computer costs if the different elements (system level, disciplines) can use surrogate models from other elements to approximately solve coordination problems. This strategy has been applied with varying success to CO, CSSO, and BLISS. For example, I. Sobieski and Kroo use response surface equations with CO [145], as do Jun *et al* [78]. Sellar, Batill and Renaud developed an RSE-based CSSO [136], [137] as well as a variant that uses artificial neural networks [135]. A later version of BLISS also uses response surface equations to aid parallel computation [147], [86], [36].

1.2.3 Limitations of Current MDO Architectures with Adaptive Learning

Despite the limited use of surrogate modeling for adaptive learning in MDO architectures, there are no methods that use local uncertainty or probabilistic information as explicit sampling criteria. As mentioned earlier, there do exist Kriging-based Bayesian adaptive sampling methods, but these have been applied only to single functions or single-level, all-at-once optimization.

If the Kriging-based adaptive sampling methods could be combined with multi-level MDO architectures, the combined framework offers a chance of computational savings for either design space exploration or optimization. It could benefit from the specialized features of the MDO architectures (e.g. disciplinary autonomy, easy distributed computing) with the balanced exploration and exploitation properties of Bayesian adaptive sampling. In a multidisciplinary design problem, each discipline could exploit and explore — not to find regions favorable to itself, but rather to aid the system-level goal.

This can be viewed as a gap in the present literature, whether from the viewpoint of model-enhanced MDO or MDO-enhanced adaptive sampling. Again, the main obstacle has likely been computational cost of the Kriging-based Bayesian models as well as a tendency of Kriging to overfit 'noisy' features in analysis functions.

In summary, current multidisciplinary design techniques partially alleviate the FPI problem but face the following difficulties:

1. DoE and surrogate modeling can waste valuable function calls in unfavorable regions of the design space. This can be partially alleviated by adaptive learning models.

2. Some multi-level MDO architectures have already incorporated adaptive learning and surrogate models. However, these algorithms do not directly exploit probabilistic measures of model uncertainty when selecting each new sample

point.

3. Some Kriging-based optimizers use Bayesian adaptive sampling for single functions. These methods have not been used in multidisciplinary architectures to exploit the structure of MDO problems.

These issues are especially problematic for computationally expensive codes. A new combined method that answers these challenges may show more information about the global optimum for a given number of function calls. This result is not as useful as an accurate surrogate model or a converged optimal solution, but it is a more sensible approach when there is no hope of either.

The proposed Bayesian collaborative sampling (BCS) method addresses these problems and builds on current methods in two ways. First, following the example of architectures like BLISS and CSSO, surrogate models are integrated into an MDO architecture to allow adaptive learning. While this general idea is not new, BCS is different because it uses fast Bayesian surrogate models. This is not a mere substitution of one type of model for another, for it allows certain types of statistical inference that are unavailable to other methods. The theory and use of Bayesian models are explained in the following theory chapters.

Second, methods of Bayesian inference are used to guide the adaptive refinement of surrogate models. The proposed method uses the predictive distributions of responses to calculate quantities like expected improvement and target matching probabilities of responses. It will be shown that these quantities from Bayesian inference can guide the adaptive learning in an MDO architecture more effectively than current methods. These techniques are relatively new; they have been used in single function optimization but have not been incorporated into MDO architectures.

1.3 Informal Preview of Bayesian Collaborative Sampling

At this point, it may be helpful to give a rough preview of Bayesian collaborative sampling. Even though a more thorough explanation will follow, it is useful to understand how the general strategy addresses the motivations.

This loose explanation is best done by analogy. The original collaborative optimization (CO), like many other bi-level methods, has a system level "boss" optimizer that sets goals for the lower level "employee" discipline optimizers. Imagine that this boss writes a set of goals on a chalkboard every morning. The employees sit at their respective cubicles, run their analyses and optimizers and try to match the chalkboard goals. After a while, they each report a single number that summarizes how well they met the targets, as well as any numbers that the boss needs to calculate the overall system objective. The boss then considers the employees' reports and writes a new set of goals on the chalkboard on the following day, hoping that the employees can match them and also compute an improved system objective value. The boss's problem is to manipulate these chalkboard goals every day until a feasible optimum is found.

In BCS, the boss and employees are each equipped with Bayesian models, which are analogous to their memories or mental impressions based on the results they have seen up to the present. Each time these people encounter new data or information, they update their mental impressions.

As in the original CO, the boss writes goals on a chalkboard. However, the employees do not immediately rush to their cubicles to run analyses and try to match the chalkboard. Rather, they contemplate their memories (Bayesian models). Without running any new analyses, they can infer from previous experiences and make statements like:

"Based on similar experiences, I think I can very likely match the chalkboard goal."

"I probably cannot match it."

"I am not sure and must run additional analyses to be certain."

With Bayesian models, they can even give number values for the probability of matching the chalkboard goals.

The boss hears these informal responses and considers his/her own system Bayesian model. Without explicitly knowing the system objective value for the current chalkboard goal, the boss can still have a rough idea of how it will compare with previous experiences. Using this guess as well as the employees' informal statements, the boss may choose whether some chalkboard goal is worth actually pursuing. If the employees are very unlikely to match, or if the expectation of system improvement is very low, then the goal settings are discarded without running any new analyses. On the other hand, if the goal settings seem sufficiently promising, then the boss directs the employees to proceed as in the original CO formulation. All results of each iteration are used to sequentially update the different Bayesian models.

Major cost savings can result from this formulation since

- Effort is not wasted to meet goals that are probably not worth pursuing, and
- Worthy goals are not pursued if they are probably impossible to meet.

As BCS progresses, the employees' experiences/impressions (disciplinary Bayesian models) become strongly focused on regions of the design space in which the boss (system optimizer) is most interested *and* where the employees are most able to meet the chalkboard settings.

The following chapter introduces theoretical background before giving a formal description of Bayesian collaborative sampling.

CHAPTER II

BRIEF OVERVIEW OF THEORY

A review of theory and current literature is given before describing the specific workings of BCS. The discussion is qualitative for now, and detailed mathematical derivations are delayed until later chapters. Four key theoretical concepts are reviewed:

1. Bayesian models and inference
2. Expected improvement
3. Target matching probability
4. Collaborative optimization

2.1 Bayesian Modeling and Inference

Bayesian modeling and inference do not refer to a specific type of surrogate model (e.g. kriging, single-layer perceptron) but rather to ways of interpreting the models and their uncertainty. Bayesian modeling and inference are best explained in terms of their difference from the classical, frequentist approach to probability and statistics. The frequentist view interprets probability as the long-term, relative frequency of some event. The Bayesian view interprets probability as a quantification of the degree of knowledge or subjective certainty [20], [11]. The difference between these paradigms has practical consequences for the BCS architecture. For simplicity, the concepts are first explained in terms of a simple linear regression model. Notation follows the conventions of machine learning and statistical pattern recognition literature, such as texts by Bishop [19] or Duda, Hart, and Stork [47].

Suppose that $t = f(\mathbf{x})$ is an unknown true response to a design point $\mathbf{x}$. The input (or design) variable $\mathbf{x}$ has dimension M, where each of the M elements represents a design attribute (wingspan, aspect ratio, etc.). The response $t = f(\mathbf{x})$ is evaluated at N different design points to collect a response sample $\mathbf{T}$. For example, a single design be represented as

$$\mathbf{x} = \begin{bmatrix} x_1, & x_2, & x_3, \cdots, & x_M \end{bmatrix},$$

and a set of N design points is written as

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \vdots \\ \mathbf{x}^{(N)} \end{bmatrix}$$

The set of response values is

$$\mathbf{T} = \begin{bmatrix} t\left(\mathbf{x}^{(1)}\right) \\ t\left(\mathbf{x}^{(2)}\right) \\ \vdots \\ t\left(\mathbf{x}^{(N)}\right) \end{bmatrix} = \begin{bmatrix} t\left(x_1^{(1)}, x_2^{(1)}, \cdots, x_M^{(1)}\right) \\ t\left(x_1^{(2)}, x_2^{(2)}, \cdots, x_M^{(2)}\right) \\ \vdots \\ t\left(x_1^{(N)}, x_2^{(N)}, \cdots, x_M^{(N)}\right) \end{bmatrix}$$

For convenience, consider $\mathbf{D} = \{\mathbf{X}, \mathbf{T}\}$ to be one "data set." Also for simplicity, assume that the number of design variables $M = 1$ for now, so each design $\mathbf{x}$ is a scalar.

A surrogate model form is then selected. In this example, consider the simple

linear model

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi} = \begin{bmatrix} w_1 \\ \vdots \\ w_L \end{bmatrix}^T \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_L \end{bmatrix} = w_1\phi_1(\mathbf{x}) + \cdots + w_L\phi_L(\mathbf{x})$$

where $\mathbf{w}$ is a vector of model parameters (weighting coefficients in this case), and $\boldsymbol{\phi}$ is a vector of L basis functions.[1] The model output, y, then, is an *approximation* of the true response t. That is,

$$y(\mathbf{x}, \mathbf{w}) = w_1\phi_1 + \cdots + w_L\phi_L \approx t(\mathbf{x}),$$

which estimates the response for a single design $\mathbf{x}$. For example, if

$$\mathbf{x} = x \quad \text{and} \quad \boldsymbol{\phi} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_L \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ \vdots \\ x^{L-1} \end{bmatrix},$$

then y becomes the familiar polynomial curve fit. Note that the model is linear with respect to $\mathbf{w}$, rather than $\boldsymbol{\phi}$, so this may still be called a linear regression despite nonlinear basis functions in $\boldsymbol{\phi}$.

In the frequentist approach, the model may be fit to the training data by finding a maximum likelihood estimate (MLE) of $\mathbf{w}$ or by minimizing an error function, as in the common least squares method:

$$E(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} [y(\mathbf{x}^{(i)}, \mathbf{w}) - t^{(i)}]^2 \tag{1}$$

One particular parameter estimate $\hat{\mathbf{w}}$ minimizes $E(\mathbf{w})$, in which case the best fit model for the given data becomes:

$$y(\mathbf{x}, \hat{\mathbf{w}}) = \hat{w}_1\phi_1 + \cdots + \hat{w}_L\phi_L \approx t(\mathbf{x})$$

[1] Please note the typographical distinction between boldface vector $\boldsymbol{\phi}$ and its elements ϕ_i, as the fonts are unfortunately difficult to distinguish in LaTeX: $\boldsymbol{\phi}\phi\boldsymbol{\phi}\phi$.

Now, given a new candidate design $\mathbf{x}^*$, the model can be used to predict the response:

$$y^* = y(\mathbf{x}^*, \hat{\mathbf{w}}) \approx t(\mathbf{x}^*)$$

A key problem is that there is no direct way to infer "true" accuracy of $\widehat{\mathbf{w}}$ or its resulting model $y(\mathbf{x}, \widehat{\mathbf{w}})$ based only on observed data $\mathbf{D}$. The frequentist view assumes that there exists some $\mathbf{w}_{\text{ideal}}$ such that $t(\mathbf{x}) = y(\mathbf{x}, \mathbf{w}_{\text{ideal}}) + \epsilon$, where ϵ is a random error with some assumed probability distribution. The principal challenge of this approach is that it is difficult to infer how well $\mathbf{w}_{\text{ideal}}$ is approximated by $\hat{\mathbf{w}}$, which is based upon only one particular data set $\mathbf{D}$ from an infinite number of possible realizations of $\mathbf{D}$. If another instance of $\mathbf{D}$ were picked, then $\widehat{\mathbf{w}}$ might be different. The frequentist paradigm tends to focus on finding the asymptotic properties of $\widehat{\mathbf{w}}$ as sample size N becomes large. Furthermore, many measures of uncertainty in the model — such as confidence intervals, prediction intervals, error bars, p-values — are based upon an imagined distribution of many possible data sets $\mathbf{D}$.

The Bayesian view is fundamentally different. First, the observed $\mathbf{D}$ is the only reality or "true" data. Second, there is no ideal, true value for $\mathbf{w}$. Rather, it is treated as a random variable. A probability distribution on $\mathbf{w}$ characterizes the uncertainty of the model parameters, given the observed data [11]. It is worth emphasizing again that the probability distribution does not represent a frequency distribution (or histogram) of $\mathbf{w}$ taking different values but rather the degree of certainty of $\mathbf{w}$ taking those values. This is called the posterior distribution over $\mathbf{w}$ given $\mathbf{D}$, written as $p(\mathbf{w}|\mathbf{D})$. The calculation of $p(\mathbf{w}|\mathbf{D})$ invokes Bayes's theorem, a prior distribution on $\mathbf{w}$, and the sum and product rules of probability; a full derivation will be given in a later chapter. For now, it will be enough to say that once it is known, the *(posterior) predictive distribution* for the response t^* at a new input point $\mathbf{x}^*$ can be calculated. For example, given a new aircraft design that has not been analyzed, one can predict a distribution on its yet unseen drag.

This predictive distribution is the key machinery of the BCS architecture. Instead

of referring to data set $\mathbf{D}$, return to the notation of sample inputs $\mathbf{X}$ and responses $\mathbf{T}$ that comprise data set $\mathbf{D}$. After observing $\mathbf{X}$ and $\mathbf{T}$, the posterior distribution over model parameters $p(\mathbf{w}|\mathbf{X}, \mathbf{T})$ can be found. Using the sum and product rules of probability, $p(\mathbf{w}|\mathbf{X}, \mathbf{T})$ can then be used to compute $p(t^*|\mathbf{x}^*, \mathbf{X}, \mathbf{T})$, where $\mathbf{x}^*$ is a new candidate design and t^* is its response. In certain cases, this predictive distribution can be found analytically. Figure 6 below shows a cartoon of predictive distributions for three different values of $\mathbf{x}^*$, with probability density shown in a pseudo-horizontal direction. The four points represent the sample data set.

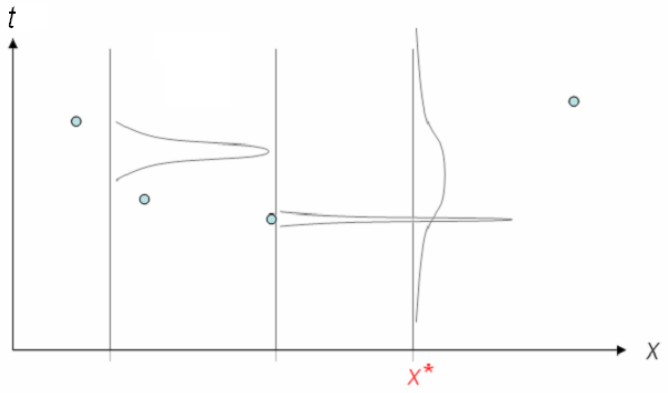

Figure 6: Predictive distributions from a Bayesian model

Summary of Frequentist vs. Bayesian Surrogate Modeling

Given a data set, the frequentist approach to surrogate modeling will give a predictive value for t^* given a new point $\mathbf{x}^*$. It addresses uncertainty of this prediction by imagining many other data sets based on an assumed noise distribution. The Bayesian approach considers just the observed data set and expresses uncertainty in terms of a posterior distribution on model parameters. This distribution can be in turn used to approximate a predictive distribution over t^* for a new input $\mathbf{x}^*$.

Example of a predictive distribution:

Let $\mathbf{x}^*$ represent an airfoil geometry and t^* represent its maximum lift to drag ratio. A training input set $\mathbf{X}$ of 500 airfoils is analyzed to find a set $\mathbf{T}$ of corresponding L/D_{max}. A Bayesian model is fit to $\mathbf{X}$ and $\mathbf{T}$. Given a new airfoil design $\mathbf{x}^*$, the model can give a predictive distribution, such as: "L/D_{max} for some particular $\mathbf{x}^*$ is predicted to have a Gaussian distribution with a mean of 14 and a variance of 3."

2.1.1 Bayesian Models: Current Literature

Bayesian models are already being used in engineering design. Efforts in that field have focused almost exclusively on Kriging—a type of Bayesian Gaussian process model that is also called Design and Analysis of Computer Experiments (DACE). Because it has been described in several recent works, a mathematical description is not given here. Although Kriging was originally developed for geostatistics, the pioneering work on its use in computer experiments was done by Sacks and Welch [128], Rasmussen and Williams [120], and later by O'Hagan, Kennedy and Oakley [108], [83]. Kriging can be interpreted as a Bayesian model for deterministic computer experiments. Often, only the mean of the predictive distribution is used to predict a response, in which case Kriging is similar to any other response surface method, as discussed by Simpson and Peplinski [142]. Kriging has been used in MDO architectures in this simple form, as Kim *et al* did with BLISS [84]. But several

studies have directly used the predictive distribution to guide sampling. These include Nair, Choudhury and Keane [104] and expected improvement-based methods discussed later.

Kriging has also been used for adaptive sampling in multi-fidelity problems. Typically, it is used to model a multiplicative or additive correction between a low and high fidelity code. At each iteration, the point of maximum uncertainty or variance is used to guide high-fidelity analysis. High fidelity training is called only in regions where a correction is most needed. This form of Kriging-based adaptive sampling has been studied intensely in recent years by Keane and associates at the University of Southampton [79], [82] as well as Gano *et al* [56], [57], [114], Huang *et al* [67], Rajnaryan *et al* [118], and Rodriguez *et al* [124]. Other research studies Kriging-based adaptive sampling in other tasks, such as multi-objective optimization [80].

The chief difficulty with Kriging is its computational expense. For even a simple single-variable function like $y = sin(x)$, fitting a Kriging model to a sample of 5000 points may take several hours of wall clock time on a typical 2009 desktop computer using statistical software like JMP®. This is because maximum likelihood estimation during model fitting requires repeated factorizations (Cholesky, LU, etc.) of a matrix whose size is the number of sample points. Nair, Choudhury and Keane describe this more specifically: if N is the number of training points, the cost of maximum likelihood estimation of hyperparameters grows as $O(N^3)$ while that of posterior predictive distributions grows as $O(N^2)$ [104]. Other authors have noted this very high cost. For example, in 2004, Leary *et al* discuss a Kriging variant with adjoint CFD analysis and recommend using it when $N(k+1)/2 < 1000$, where N is the sample size and k is the number of design variables [90].

For a one-time modeling of a DoE sample, the computation cost may not be a great problem. But a adaptive learning algorithm requires repeated re-fitting of models—one for each iteration. Toal *et al* (2008) also note the high cost of Kriging sequential

updates for optimization and studied the performance of different hyperparameter tuning strategies to reduce this cost [151]. For a modest engineering problem of $O(10)$ variables and $O(1000)$ sample points, the cost of Kriging updates may be far more expensive than the high fidelity function calls that it is supposed to save.

In addition to speed issues, Bayesian adaptive learning with Kriging often encounters numerical problems as the algorithm converges or in 'noisy,' highly multimodal problems. These issues will be discussed in later chapters. Some efforts have been made to modify Kriging-based learning for noisy experiments by regularization [54]. However, these problems may also be avoided by Bayesian learning with alternate Bayesian models.

The fields of machine learning and pattern recognition have made many advances in Bayesian learning. Typically, researchers in this field develop Bayesian models for regression or classification problems. Adaptive learning algorithms often accept a stream of uncontrolled data and uncover an underlying function or discover a surface that separates classes of data. Examples include adaptive spam filters and speech or handwriting recognition. These problems are different from the present concern, which is a goal-oriented, supervised learning where the experimenter selects each input point. However, many of the insights from machine learning can be useful for design optimization. In particular, much research in the past twenty years has focused on a class of regression and classification models called sparse kernel methods, such as the Support Vector Machine (SVM) [154], [35], [64]. The SVM is a popular method in the machine learning community because of its generalization abilities and speed due to its sparse set of basis functions. These advantages recently led some aerospace design researchers to explore the use SVM as a surrogate modeling method, including Ran and Mavris [119], Fan and Dulikravich [48], Clarke [31], and Forrester, Keane and Sobester[52]. More recently still, Bayesian variants of the sparse kernel machines have emerged, such as the Relevance Vector Machine (RVM) [150]

[19]. RVMs have a functional form similar to SVMs but are Bayesian models that yield predictive distributions rather than just response predictions. This thesis uses sparse Bayesian models, which are a generalization of RVMs that can use non-kernel basis functions. Sparse Bayesian models and RVMs use a mechanism called automatic relevance determination (ARD) to prune unnecessary basis functions from the model, which leads to greater speed in required for adaptive model fitting. This speed does incur some serious costs, which are discussed in the next chapter.

Having summarized Bayesian models and predictive distributions, the discussion turns their use in Bayesian adaptive learning.

2.2 Expected Improvement

Expected improvement is a criterion for adaptive learning with Bayesian models, and it is one of the main elements that enable BCS. The idea was developed in the context of gradient-free optimization of blackbox functions with probabilistic surrogate models. The general approach is to use the posterior predictive distributions on the response to show the best new point on which to conduct the next analysis, given the current observed data.

There are many variations on expected improvement, including Mockus [102], Cox and John [33], and Jones, Schonlau, and Welch [77]. This thesis will use a version recently described by Forrester, Keane and Sobester [52]. Consider the example problem of minimizing $t(\mathbf{x})$ with respect to $\mathbf{x}$, which has one dimension in this example. Assume that there are N sample points that have already been evaluated by the analysis code, $t(\mathbf{x})$. These are represented by the four points in Figure 7 below. One of the points has the best (lowest) response value, t_{best}.

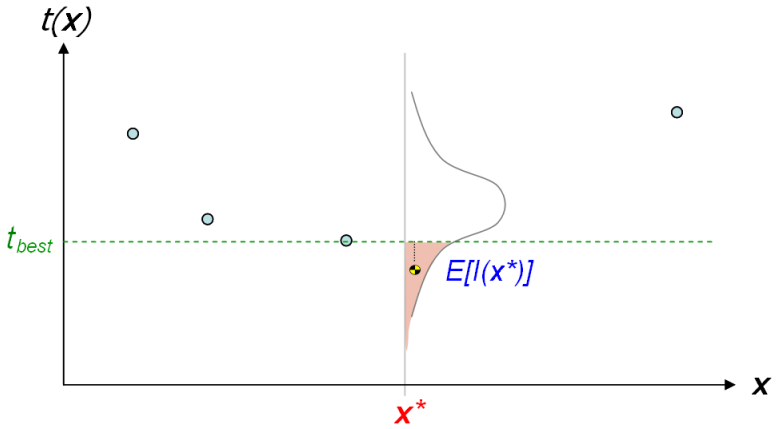

Figure 7: Expected improvement

Suppose that a Bayesian model has been fit to this training sample of four inputs and outputs, denoted $\mathbf{X} = \left[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}, \mathbf{x}^{(4)}\right]$ and $\mathbf{T} = \left[t^{(1)}, t^{(2)}, t^{(3)}, t^{(4)}\right]$. At any arbitrary new design point $\mathbf{x}^*$, the Bayesian model gives a posterior predictive distribution $p\left(t^* | \mathbf{x}^*, \mathbf{X}, \mathbf{T}\right)$. This distribution is shown in the pseudo-horizontal direction

above. Part of the distribution is lower than the current best response, t_{best}. The value of the cumulative distribution function at t_{best} is represented by the shaded area. The fraction of this area compared to the area of the entire distribution represents the probability that $t(\mathbf{x}^*)$ is an improvement over the current best response, which is denoted $P\left[I(\mathbf{x}^*)\right]$. The first moment of the probability of improvement over t_{best} is the expectation of improvement, $E\left[I(\mathbf{x}^*)\right]$.

Expected improvement can be used as a sampling criterion for adaptive learning. In other words, after a Bayesian surrogate model is fit to a sample of N points, the point of maximum expected improvement can be used to guide the $(N+1)$th sample location. $E\left[I(\mathbf{x}^*)\right]$ can be very quickly calculated from a Bayesian surrogate model, so the design space can be searched for its maximum using a genetic algorithm or similar optimizer. After the new point is evaluated for its true response, the model is fit again with $N+1$ points, and the process continues. This process is shown in a notional illustration in Figure 8 and Figure 9 below.

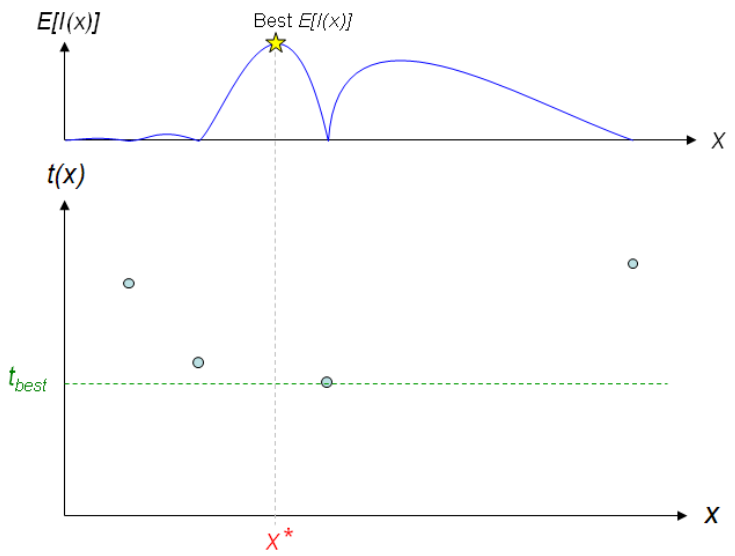

Figure 8: Adaptive learning by expected improvement: Step 1

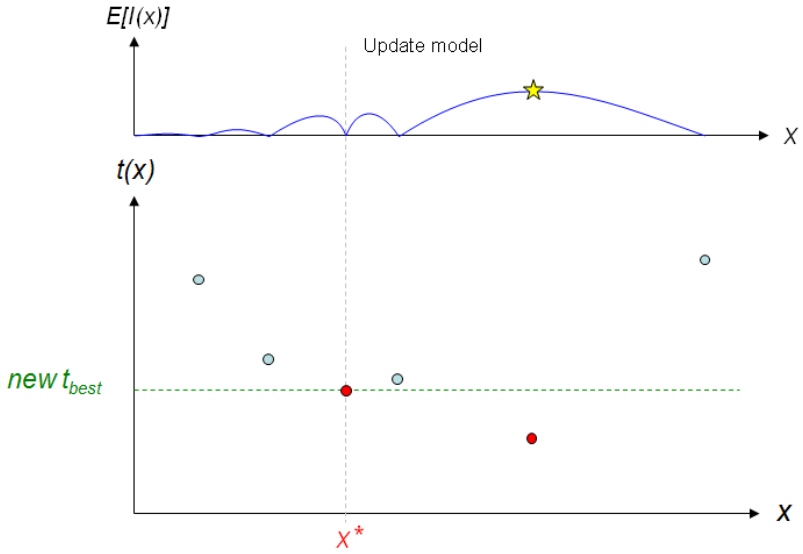

Figure 9: Adaptive learning by expected improvement: Step 2

First, in Figure 8, the four data points are used to regress a Bayesian model. The predictive distribution is used to calculate $E\left[I(\mathbf{x}^*)\right]$. An optimizer finds the point of maximum $E\left[I(\mathbf{x}^*)\right]$ (star). This point, $\mathbf{x}^*$, is evaluated (darkened point).

Then, in Figure 9, the Bayesian model is updated with the new point. Since this

new point has the lowest response value, it sets the new t_{best}. $E\left[I(\mathbf{x}^*)\right]$ assesses improvement over the new t_{best}. Another maximum $E\left[I(\mathbf{x}^*)\right]$ is found, and the adaptive learning proceeds.

The maximum $E\left[I(\mathbf{x}^*)\right]$ points in the two steps above demonstrate the advantage of sampling by expected improvement. $E\left[I(\mathbf{x}^*)\right]$ can be high for two reasons: 1) there is a general trend in the data such that the mean of the predictive distribution is lower than t_{best}, or 2) the distribution is uncertain and spread widely so that a large portion lies below t_{best}. In Figure 9, the second reason seems to dominate, since the maximum $E\left[I(\mathbf{x}^*)\right]$ occurs in a sparsely sampled region. These two mechanisms tend to automatically balance between exploitation of known regions and exploration of unknown regions. The balance makes it less likely that sampling scheme will focus on a deceptive local minimum region without adequately investigating other regions.

In the past ten years, there have been many applications of this method. For example, Glaz, Friedmann and Liu apply it under Jones, Schonlau and Welch's "Efficient Global Optimization" name to helicopter rotor blade design [59]. Jeong *et al* apply it to two dimensional airfoil design [73], and Todoroki *et al* use it to design hat-stiffened composite panels [152]. Keane further develops the expected improvement concept to find a Pareto front in a multiobjective design problem [80].

2.3 Goal Seeking and Probability of Target Match

In addition to expected improvement, BCS also uses a adaptive learning criterion based on the probability that target response value, t_{target}, could exist at some point $\mathbf{x}^*$. This is shown notionally in Figure 10 below.

In practice, the sampling criterion could be computed by

$$P(t_{target} - \epsilon < t < t_{target} + \epsilon) \tag{2}$$

for some small ϵ.

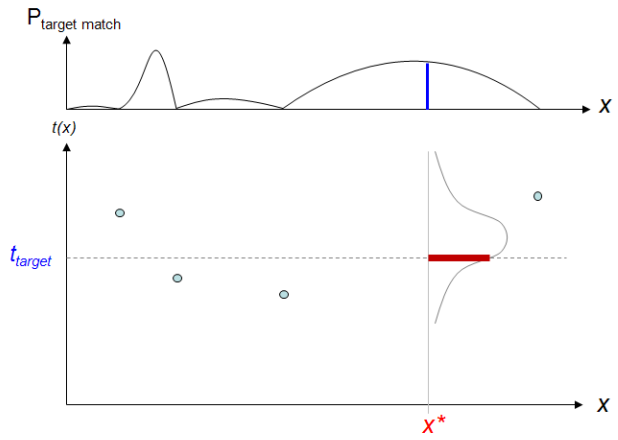

Figure 10: Goal seeking adaptive learning by probability of target match

Adaptive learning based on this goal seeking/target matching criterion will tend to cluster sample points and increase accuracy in locations where t_{target} is likely. As with expected improvement, it will also tend to balance between exploitation of the known and exploration of the unknown. Forrester *et al* describe a similar *conditional likelihood* criterion for Kriging models in [52]. It hypothesizes a target response value at some new point $\mathbf{x}^*$, uses this value in likelihood function that is conditional on observed data, uses maximizes conditional log-likelihood to identify the next sample point. Details of this method are not repeated here, but the role is similar to Figure 10.

To summarize Bayesian inference and the adaptive learning techniques, Figure 11 demonstrates the two learning criteria in the minimization of a typical test function. The two input variables are on the horizontal axes. In Figure 11, the top left shows a modified Rastrigin test function directly evaluated at 70 x 70 grid points. At top right, a Latin hypercube DoE of 50 points is evaluated. A relevance vector machine (RVM) is fit to the sample, giving predictive distributions on response t. At bottom left, this RVM Bayesian model is used to calculate $-\log E[I(x)]$ based on the current best, $t_{best} = 2.2$. The negative sign is used to put the adaptive learning problem in the form of a minimization to follow the convention of optimization problems. The high plateaus correspond to $-\log E[I(x)]$ near zero, where improvement is unlikely. In

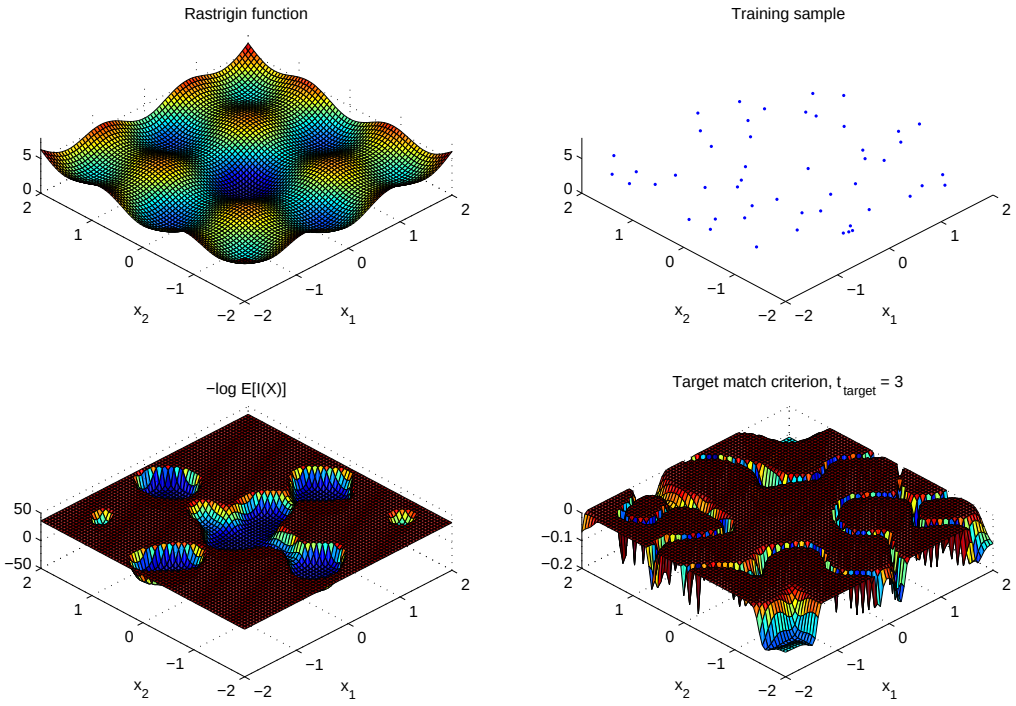

Figure 11: Modified Rastrigin function example

an adaptive learning process, these plateau regions would be ruled out from further analysis. Note that the plateaus can correspond to either isolated sample points with high responses or densely sampled regions where responses are even slightly worse than the current best. Bottom right is the target matching case, showing the negative probability of the response being within some threshold of the target,

$$-P(t_{target} - \epsilon < t < t_{target} + \epsilon),$$

where $t_{target} = 3$ and $\epsilon = 10^{-2}$. The lower trench-like regions are more likely to match the target.

These adaptive sampling criteria will be shown in greater mathematical detail in Chapter 3. Having briefly described some Bayesian adaptive learning techniques, attention is shifted to MDO architectures which will incorporate them. There is no obvious connection between these areas, but they will combine in the final BCS

formulation.

2.4 Overview of MDO Architectures

Recall that the goal of this thesis is to combine the adaptive learning Bayesian models with a bi-level MDO architecture, so that disciplinary learning is coordinated toward a system level optimum while preserving disciplinary autonomy. Recent MDO literature includes at least four multi-level architectures that might be adapted for this purpose: collaborative optimization (CO), concurrent subspace optimization (CSSO), bi-level system synthesis (BLISS), and analytic target cascading (ATC). Among these and other architectures, CO is chosen as a basis for the present method because of its way of enforcing interdisciplinary compatibility (or "consistency"). This feature makes CO well suited to the previously discussed Bayesian adaptive learning techniques. It may be possible to craft Bayesian adaptive variants of CSSO, BLISS, or ATC, and this should be considered for future research. But for now, there is a natural and intuitive fit between Bayesian methods and CO. So, this combination is chosen as a first, proof-of-concept multidisciplinary sampling scheme.

First, some background on interdisciplinary coupling is given. Then, the original CO is described. This finally leads to the BCS architecture.

2.4.1 Coupling Variables and Interdisciplinary Compatibility

When multidisciplinary analysis (MDA) is done on a single candidate design, there are often intermediate variables communicated between the disciplines. These may be distinct from the actual design variables. These coupling variables (or linking variables) are a key reason for selecting CO as the framework for adaptive learning. Consider a design structure matrix (DSM or N^2 diagram) of an example multidisciplinary problem as shown in Figure 12.

Each contributing analysis (CA) represents a discipline - in this case, aerodynamics and structures. Connecting lines above the diagonal represent forward information

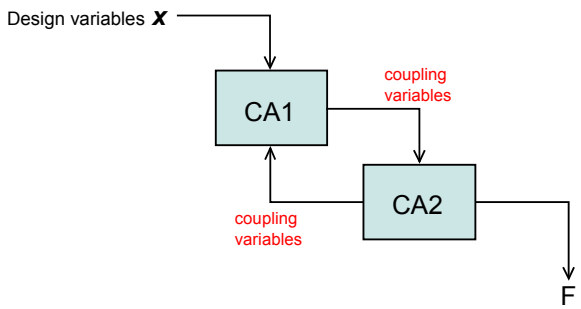

Figure 12: Example design structure matrix

flow, while connecting lines below the diagonal represent feedback. In this example, there are two contributing analyses and loop between from CA2 to CA1. The problem with this loop is that the coupling variables passed from CA1 to CA2 depends on the coupling variables passed back from CA2 to CA1. Values for the coupling variables must be consistent for both CAs to achieve interdisciplinary compatibility. Therefore, to evaluate a single design, the loop must be repeated until the coupling variables converge. As described earlier, this is a fixed point iteration (FPI), where relations of the form

$$\mathbf{x} = f(\mathbf{x})$$

must be solved. Because of this FPI, several CA function calls are required to evaluate the overall objective function $F(\mathbf{x})$ for a single design $\mathbf{x}$. This is a very costly method for achieving interdisciplinary compatibility during design analysis, especially if the CAs are expensive codes like CFD or finite element method (FEM).

For a more concrete example, consider the problem in Figure 13.
In this case, the coupling variables between the aerodynamics and structures CAs are load distribution, deformation, and wing weight. The structures CA must provide both the wing weight and deformation to the aerodynamics CA, but it cannot do this until external loads are provided by the aerodynamics CA. Aerodynamic analysis cannot be done without knowing a wing weight because the analysis must correspond to steady level flight for a known total aircraft weight. Also, aerodynamic analysis

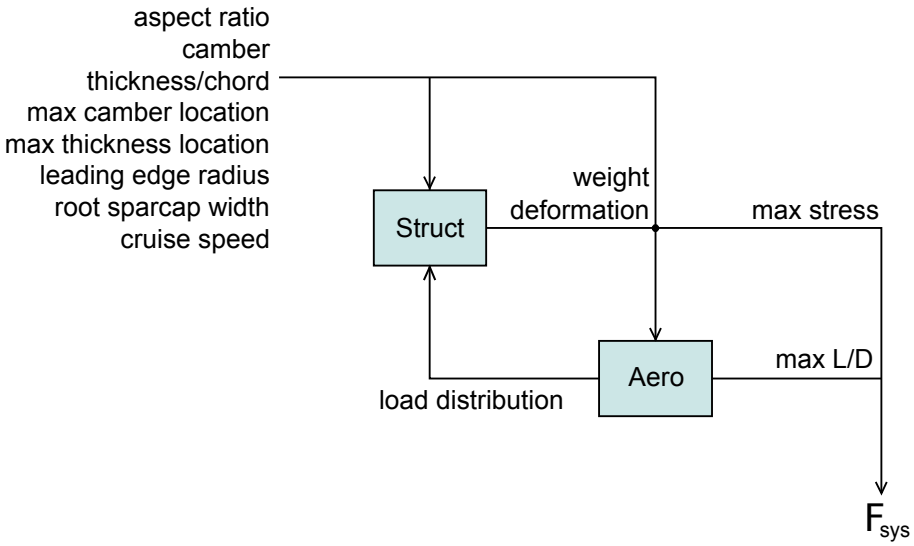

Figure 13: Example problem: Aero-structural optimization of a wing

must be done on the deformed, in-flight shape of the wing rather than its unloaded jig shape. To break this impasse, the CAs must make an initial guess at the coupling variables and conduct fixed point iteration until those variables converge.

Recall that the discussion so far only concerns a single evaluation of the objective function. In a direct (all-at-once) optimization problem, the entire DSM could be a blackbox function, which would then be controlled by an external optimizer.

2.4.2 Collaborative Optimization

Decomposition of the MDO problem can alleviate the cost of interdisciplinary consistency in multidisciplinary analysis. CO modifies the original problem into a bi-level optimization. The optimization is split into a system level and discipline (subspace) level optimizers, as shown in Figure 14.

The system optimizer sets targets for coupling variables and global/shared variables. The disciplinary subspace consists of local variables that concern only the discipline as well as local copies of the target variables. The disciplinary optimizers manipulate these variables to minimize the discrepancy between local copies of the targets and the actual targets. The system optimizer searches target settings to

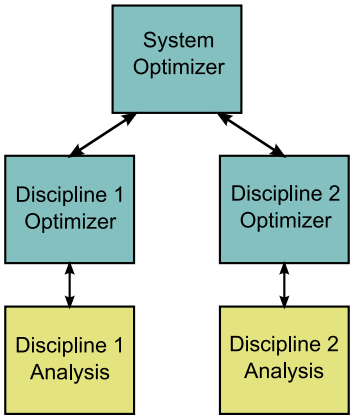

Figure 14: General structure of collaborative optimization

minimize the main objective function.

Returning to the wing example, imagine that an engineering firm has a system-level "boss" and disciplinary teams for aerodynamics and for structures. At each system-level iteration, the boss writes values for target variables on a chalkboard. These target variables include the coupling variables (load distribution, wing weight, and deformation) and global/shared variables (wing planform and thickness). The aerodynamics team trusts that the structures team will somehow match the weight and deformation goals written on the boss's chalkboard. The aerodynamics team then manipulates local aerodynamic variables (airfoil geometry) in order to match the load distribution target on the chalkboard. If it cannot do this with airfoil geometry alone, it also manipulates local copies of the target variables to gain extra degrees of freedom. The discrepancy between actual targets and local copies of the targets is minimized.

Both disciplinary teams perform such an optimization to match the chalkboard targets. Each team reports to the boss how well they could match the targets - that is, the amount of discrepancy between the targets and their local copies. If both teams report zero discrepancy, they achieve interdisciplinary compatibility. The teams also report any quantities required for system level objectives (drag) or constraints (maximum stress, deflection). The boss considers this information and writes new targets on the chalkboard. The boss must select the targets in a way that minimizes drag —

subject to the constraint that the teams can match them with zero discrepancy.

Note that the disciplinary teams do not communicate with each other, and each has autonomy to choose its own optimization methods. Yet, the teams collaborate toward interdisciplinary consistency by working toward common chalkboard targets.

There are several versions of CO since its original proposal by Braun [22], [23]. These include an RSE-equipped CO by Sobieski and Kroo [145], "enhanced CO" by Roth and Kroo [127], and "robust CO" by Gu and Renaud [62]. The following notation is slightly modified from Braun [23]. First, the original optimization problem is stated as a general, non-linear programming (NLP) problem:

$$\begin{aligned} &\text{Minimize:} && F(\mathbf{x}) \\ &\text{with respect to:} && \mathbf{x} \\ &\text{subject to:} && G(\mathbf{x}) \leq 0 \\ & && H(\mathbf{x}) = 0 \end{aligned}$$

where G and H are miscellaneous constraints. CO modifies this into a non-convex, bi-level NLP:

Collaborative Optimization

Assume there are I disciplinary subspaces. Design variable X is decomposed into shared, coupling, and local variables of the I subspaces:

$$\mathbf{X} = \left[\mathbf{x}_{\text{shared}}, \mathbf{x}_{\text{coupling}}, \left(\mathbf{x}_{\text{local}}^{(1)}, \mathbf{x}_{\text{local}}^{(2)}, \ldots, \mathbf{x}_{\text{local}}^{(I)}\right)\right]^T$$

Let system target vector $\mathbf{Z} = [\mathbf{x}_{\text{shared}}, \mathbf{x}_{\text{coupling}}]^T$.

System Level Problem

$$
\begin{array}{lll}
\text{Minimize:} & F(\mathbf{Z}) & \\
\text{with respect to:} & \mathbf{Z} & \\
\text{subject to:} & Q_i(\mathbf{Z}) & = 0 \text{ for } i = 1, 2, \ldots, I \\
 & & \text{(target discrepancy of each discipline)} \\
 & G(\mathbf{Z}) & \leq 0 \text{ (miscellaneous system constraints)} \\
 & H(\mathbf{Z}) & = 0
\end{array}
$$

Discipline Level Problem

Let $\mathbf{Z}_{\text{sys}}^{(i)}$ be the elements of $\mathbf{Z}$ that are relevant to the i-th discipline.
Let $\mathbf{Z}_{\text{local copy}}^{(i)}$ be a local copy of elements of $\mathbf{Z}_{\text{sys}}^{(i)}$.
$\mathbf{Z}_{\text{sys}}^{(i)}$ from system level is treated as a fixed parameter, whereas $\mathbf{Z}_{\text{local copy}}^{(i)}$ is a disciplinary variable.

For $i = 1, 2, \ldots, I$:

$$
\begin{array}{lll}
\text{Minimize:} & Q_i\left(\mathbf{Z}_{\text{local copy}}^{(i)}, \mathbf{x}_{\text{local}}^{(i)}\right) & = \left\| \mathbf{Z}_{\text{sys}}^{(i)} - \mathbf{Z}_{\text{local copy}}^{(i)} \right\|_2^2 \\
\text{with respect to:} & \mathbf{Z}_{\text{local copy}}^{(i)},\ \mathbf{x}_{\text{local}}^{(i)} & \\
\text{returning to system level:} & Q_i & \\
 & F(\mathbf{Z}), G(\mathbf{Z}), H(\mathbf{Z}), & \text{if calculated in the } i\text{-th discipline.}
\end{array}
$$

2.5 Bayesian Collaborative Sampling

Having reviewed Bayesian adaptive learning methods and collaborative optimization, the discussion turns to the Bayesian collaborative sampling (BCS) architecture that combines all of these elements.

The formulation of CO is not significantly changed, but Bayesian inference is added. In other words, BCS poses the a similar two-level problem structure as in the Section 2.4.2. It still has closed disciplinary analysis, closed design constraints, and open interdisciplinary consistency constraints. The CO formulation requires a system and discipline level minimizations, but it does not specify any particular algorithm for minimization. The BCS method specifies a Bayesian adaptive learning to drive the CO architecture.

It is similar to the original collaborative optimization architecture, but the system and discipline levels are now equipped with Bayesian surrogate models that are adaptively updated with every new output, as in Figure 15.

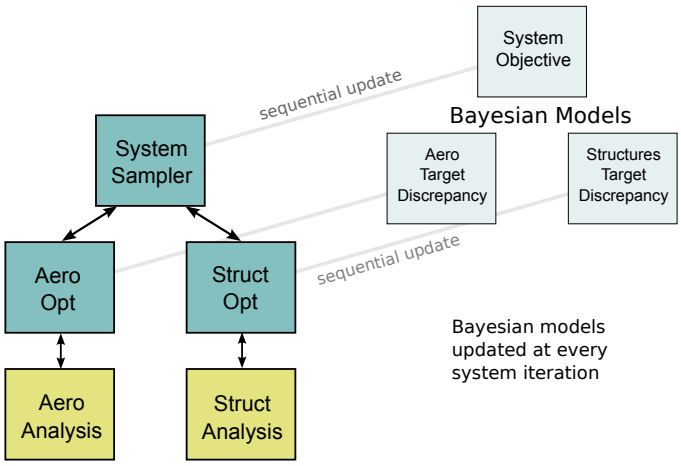

Figure 15: adaptive updating of Bayesian models in BCS

At the beginning of each system iteration, the Bayesian models are searched to find a new system design point $\mathbf{Z}_{new}$ that has the highest expected improvement in the system objective F while simultaneously maximizing the probability of interdisciplinary compatibility. This is shown in Figure 16.

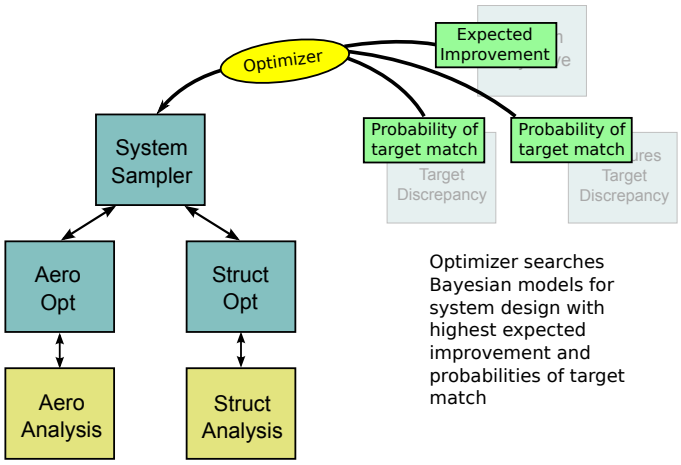

Figure 16: Point with highest expected improvement and probability of compatibility passed to system optimizer

The latter probability can be calculated by a target matching criterion. Such evaluations of expected, compatible improvement can be done very rapidly with the Bayesian models, so $\mathbf{Z}_{new}$ can be found with an optimizer. Once the best candidate $\mathbf{Z}_{new}$ is selected, this value is passed to the system optimizer, and the original CO proceeds in one system iteration.

2.5.1 A More Formal Explanation

The BCS formulation is given and explained afterward. The main difference with CO is that target discrepancy metrics are not just tracked by the Q_i for I entire disciplines, but rather individual target discrepancy metrics J_k for each of K individual coupling variables. Let system target vector $\mathbf{Z} = [\mathbf{x}_{\text{shared}}, \mathbf{x}_{\text{coupling}}]^T$.

System Level Problem

Minimize:	$F(\mathbf{Z})$	(including constraint penalties)
by sampling min. of:		$-\log E\left[I\left(F(\mathbf{Z})\right)\right] - \sum_{k}^{K} \log P\left(-\epsilon < J_k\left(\mathbf{Z}\right) < \epsilon\right)$
with respect to:	$\mathbf{Z}$	(shared design variables and coupling variables)
subject to:	$Q_i(\mathbf{Z})$	$= 0$ for $i = 1, 2, \ldots, I$
		(target discrepancy of each discipline)
	$G(\mathbf{Z})$	≤ 0
	$H(\mathbf{Z})$	$= 0$
and updating:		Bayesian models $p(F\|\mathbf{Z}), p(J_k\|\mathbf{Z})$
		for $k = 1, 2, \ldots K$ coupling variables

Discipline Level Problem

Each discipline has produces local outputs $Z_{\text{k, local output}}$ that correspond to system targets for $k = 1, 2, ..., K^{(i)}_{\text{local}}$ locally relevant coupling variables. For $i = 1, 2, \ldots, I$ disciplines:

Minimize:	$Q_i\left(\mathbf{x}^{(i)}_{\text{local}}\right)$	$= \sum_{k}^{K^{(i)}_{\text{local}}} J_k^2$ for $J_k = Z_{\text{k,sys}} - Z_{\text{k, local output}}$
with respect to:	$\mathbf{x}^{(i)}_{\text{local}}$	
returning:		Target discrepancies J_k
		$F(\mathbf{Z}), G(\mathbf{Z}), H(\mathbf{Z})$, if calculated in the i-th discipline.

2.5.1.1 Explanation of System Level

In the original CO, the system optimizer would search for the minimum $F(\mathbf{Z})$ and enforce interdisciplinary consistency $Q(\mathbf{Z})$ from each discipline. In the BCS method, the system level search is indirect; it uses an optimizer to search a Bayesian criterion to select the best new sample point rather than to trying to directly search for the optimum. This procedure uses Bayesian predictive distributions on F and on individual target discrepancies J. In the current methods, F may include penalty functions on constraints G and H, although $G(\mathbf{Z})$ and $H(\mathbf{Z})$ may be modeled separately.

At each system iteration, an optimizer searches these models for $\mathbf{Z}$ that maximizes the expectation that F (including constraint penalties) will improve—while simultaneously maximizing the probability of interdisciplinary consistency and feasibility. These quantities to be maximized are:

$$E\left[I\left(F(\mathbf{Z})\right)\right] \tag{3}$$

$$P\left(-\epsilon < J_k\left(\mathbf{Z}\right) < \epsilon\right) \text{ for } k = 1, 2, \ldots K \text{ coupling variables} \tag{4}$$

for some small ϵ. BCS may optionally use an alternate form of Eq.4 rewritten as

$$P\left(J_k^2\left(\mathbf{Z}\right) < \epsilon\right) \text{ for } k = 1, 2, \ldots K \tag{5}$$

since J^2 is always nonnegative. In practice, it may be assumed that the improvement and constraint distributions are independent. So, instead of maximizing the three quantities separately, the sum of negative logarithms are minimized for numerical convenience. The system sampling criterion can be collected in a single expression:

$$-\log E\left[I\left(F(\mathbf{Z})\right)\right] - \sum_k \log P\left(-\epsilon < J_k\left(\mathbf{Z}\right) < \epsilon\right) \tag{6}$$

After this objective is minimized with respect to $\mathbf{Z}$, the resulting "most promising" system target $\mathbf{Z}^*$ is passed to the discipline level.

2.5.1.2 Discipline Level

The discipline level is similar to the original CO with three differences. The $\mathbf{Z}^{(k)}_{\text{local copy}}$ term in CO is not directly manipulated by an optimizer in BCS — only the local design variables $\mathbf{x}^{(i)}_{\text{local}}$ are varied. However, it can alternately be included in the discipline problem with no major theoretical difference. Second, each time the disciplinary optimization is completed, individual target discrepancies J_k are passed back to the system level instead of the aggregate Q_i as in the original CO. Third, the predictive distribution from this model can be used to find $\mathbf{x}^{(i)*}_{\text{local}}$ that minimizes $P(-\epsilon < J_k < \epsilon)$ for the k values relevant to discipline i, and this point can be used as an initial guess for the disciplinary optimization. This last technique is optional. In preliminary experiments, its benefit over the original CO disciplinary optimization appears to depend heavily on the optimizer and disciplinary function.

The BCS process can be "warm-started" by evaluating an initial sparse DoE of the system target variables $\mathbf{Z}$ for F and Js. The warm-start sample size is discussed in later chapters. After this initial DoE, Bayesian surrogate models are fit to the data sets.

In practice, the system level step of minimizing

$$-\log E\left[I\left(F(\mathbf{Z})\right)\right] - \sum_k \log P\left(\epsilon < J_k\left(\mathbf{Z}\right) < \epsilon\right) \tag{7}$$

can be done by a global optimizer such as a genetic algorithm or other metaheuristic optimizer. Note that this formulation can be slightly modified to include Bayesian modeling of constraints $G(Z)$:

$$-\log E\left[I\left(F(\mathbf{Z})\right)\right] - \sum_k \log P\left(G\left(\mathbf{Z}\right) < \epsilon\right) - \sum_k \log P\left(\epsilon < J_k\left(\mathbf{Z}\right) < \epsilon\right) \tag{8}$$

Also, note that the probability of target match in Eq.7 can alternately be expressed as

$$P\left(J_k^2\left(\mathbf{Z}\right) < \epsilon\right) \tag{9}$$

Both the alternative forms with constraint model of $G(\mathbf{Z})$ and $J_k^2(\mathbf{Z})$ were used in some early tests of BCS.

The discussion up to this point has assumed that quantities like Eq.7 can be evaluated quickly. It also assumes that the model updates at every iteration are computationally affordable so that $\text{time}_{\text{model fit}} << \text{time}_{\text{analysis}}$. The next chapter elaborates on Bayesian theory and ends with a sparse Bayesian model that is practical for this purpose.

2.5.2 Practical Implementation

The steps for a practical computer implementation of BCS are described. This consists of two distinct parts: 1) the implementation of a computer code for BCS in general and 2) its use on a particular engineering problem.

2.5.2.1 Setting Up a BCS Code

For this thesis, the entire BCS environment was created in Matlab. The general BCS code contains four major elements:

1. System analysis function that passes relevant system variables to each discipline and recovers J_k and F information

2. Disciplinary optimizers to minimize local discrepancy J_k from system targets

3. Bayesian model fitting and updating for target discrepancy J_k and system objective F

4. System optimizer to find candidate system design $\mathbf{Z}$ that minimizes the Bayesian sampling criterion

The first element, the system analysis function, essentially wraps around the disciplinary optimizers and passes inputs and receives outputs from them. Assume that the designer has decomposed a design problem into a bi-level problem for BCS. The

system analysis function assigns particular system variables to particular disciplines. For example, consider the aero-structural design of a wing. The system function is set up to pass shared/global variables such as wing thickness and aspect ratio to both aerodynamics and structures disciplines. It passes somewhat different system target variables to each discipline: it passes an aerodynamics load distribution as a target for the aerodynamics discipline, whereas it passes this load distribution as an input parameter for the structures discipline. Similarly, variables describing the deformed shape of the wing are passed to the aerodynamics discipline as inputs used for boundary conditions, but the same variables are sent as targets for structural optimization. After the disciplinary optimizers do their work, they pass target discrepancies and information for computing the objective to the system function. These discrepancies J_k and the objective function are the main outputs of the system function.

The second element in the BCS code consists of disciplinary optimizers. Each discipline receives system targets and shared design variables from the system analysis function. However, each discipline also may have local variables that only exist in that discipline. For example, the system function may pass wing area and thickness to the structures discipline, but variables like the internal spar thickness are purely structural variables that do not exist in the aerodynamics design subspace. Each discipline optimizer is arranged to minimize discrepancy from system targets, J_k, by manipulating such local variables. If there are no local variables, then the discipline problem is simply one of evaluating the system target discrepancies. In many cases, a discipline may have multiple relevant system targets. For example, the aerodynamics discipline may have to match targets for several parameters describing a load distribution. The discipline level optimization can minimize a single objective by collecting discrepancies for several targets in one aggregate criterion for the discipline, Q, which can be defined as the sum of squares of target discrepancies for the i-th discipline. Even though the optimization is done on this Q, note that it is the J_k

values rather than the minimum Q that are the main outputs that are returned to the system analysis function. The choice of optimization method depends on the problem. For example, if the disciplinary objective J_k is known to be relatively smooth and unimodal, then gradient-based optimizers may be used. However, such optimizers may show poor convergence for non-smooth, 'noisy' analysis functions that have internal solver tolerances and errors. For this thesis, a simplex-based method or a golden-section search (if performing a single variable line search) are used.

The third element is Bayesian model fitting. After each system iteration, there is a need to fit or update Bayesian models for each of the target discrepancies J_k and the objective F. The Bayesian modeling function takes an input of the current sample and responses and fits model parameters (or hyperparameters). It does not return a particular output value, but rather a fit model that allows rapid evaluation of the Bayesian sampling criteria: expected improvement and probability of target match. Note that the BCS formulation does not require a particular model form so long as the model provides Bayesian predictive distributions. In later chapters, the choice of Bayesian model implementation will be discussed in greater detail.

Fourth, a system-level optimizer is created to minimize the combined Bayesian sampling criterion (expected improvement of the objective and probability of target match) with respect to system design variables. After the Bayesian models have been fit using the third element, the Bayesian predictive quantities can be evaluated very rapidly. A genetic algorithm, for example, can be used as this optimizer. In summary, the fourth element takes current Bayesian models for J_k and F and minimizes a sampling criterion to finds the most promising system design for the next system analysis.

The four elements of the BCS code are organized as in Figure 17.

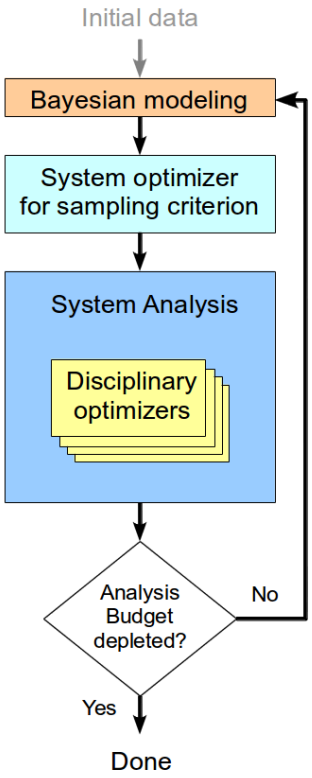

Figure 17: Generic BCS computer code requires four main elements (shaded)

2.5.2.2 Running BCS for a Particular Design Problem

Once the general computer code for BCS has been created, it must be adapted for a particular multidisciplinary engineering problem. The order of steps are described for modifying a new problem for BCS, setting up codes, providing this initial data, and then executing a BCS code.

1. **Bi-level Decomposition of Design Problem**

   (a) Restate the original multidisciplinary design problem as a bi-level formulation with system and discipline level variables

2. **Set Up Discipline Analysis Codes**

   (a) 'Wrap' and automate discipline analysis codes to accept inputs and post-process outputs required by the discipline level problem defined above

3. **Initialization**

   (a) Generate a warm-start DoE of system variables $\mathbf{Z}$

   (b) For each DoE point $\mathbf{Z}$, execute system analysis

   - Pass relevant system design variables and targets to each discipline
   - Run disciplinary optimization to minimize target discrepancies, J_k, with respect to local variables
   - Information required for the system objective is also computed

   (c) Fit Bayesian model to system objective and each target discrepancy with respect to system variables

4. **Adaptive Sampling (Execute BCS Code)**

   (a) Using an optimizer, search Bayesian models to find the most promising system design $\mathbf{Z}^*$ that minimizes the Bayesian sampling criterion

   (b) At $\mathbf{Z}^*$, execute one system analysis

   - As before, run discipline optimizers to minimize J_k
   - Compute system objective

   (c) Re-fit Bayesian models of the system objective and each target discrepancy

   (d) If computational budget is not depleted, go to Step 4 (a)

When the computational analysis budget runs out and the BCS procedure is complete, the designer is left with two products: the Bayesian models and the final sample points. The following section describes how these results can be used in a complete design process.

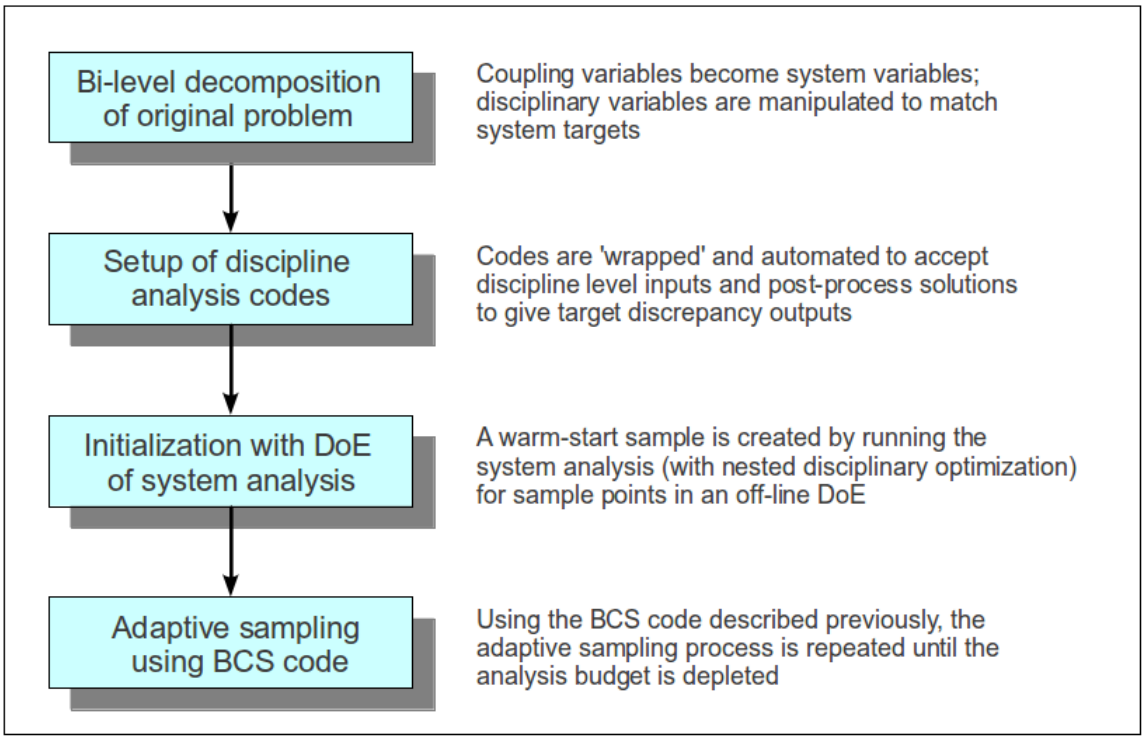

Figure 18: Procedure for setting up and running BCS for a particular problem

2.6 Use of BCS in a Design Process

BCS is not a complete design solution, but it is a tool that can aid a designer in the broader context of early design. One practical design process using BCS is described.

First, a design problem and its requirements are formulated into an objective function and constraints. Currently, BCS only addresses single objective functions; if there are multiple objectives, these must be collected into an overall evaluation criterion by a weighted sum. Then, candidate design variables and their ranges are identified. Based on the physics of the problem, the minimum fidelity levels for analysis codes are selected. Next, depending on the computational budget and the physics of the problem, the designer chooses a manageable subset of design variables. This can be done by experience, physical intuition, or statistical screening techniques such as analysis of variance (ANOVA), although this may consume some of the computational budget. After this, a design structure matrix is created, mapping the relationship

among disciplines and the flow of information.

The DSM, design variables, and analysis codes are considered together to select the design strategy. If all of the codes are relatively cheap, then established techniques of DoE and surrogate modeling can be used. Although inefficient, such methods are simple to set up and provide the designer with a global view of the design space. If the design space topology is expected to be relatively simple (smooth, unimodal, convex, etc.), then direct optimization may also be done using one of the existing MDO architectures.

However, the computer codes may be expensive because of irreducibly complex physics. Then, neither DoE nor direct optimization are practical. The designer can choose to do adaptive sampling, to get the most information out of a budget of function calls. First, he/she must examine the DSM. If the matrix is upper triangular, there are no feedbacks. Since there is need for fixed point iteration, the entire DSM can be treated as a single blackbox function. In this relatively rare case, there is no need to decompose the original problem into a new, multilevel problem. The multidisciplinary design space can be explored with existing, single-function, Bayesian adaptive sampling methods like EGO. The designer might also choose this if the computational cost is dominated by one discipline, for there may be little profit in decomposition architectures compared to fixed point iteration. However, if the codes are expensive and there is a FPI problem, then BCS is recommended.

There are at least two strategies for using BCS, although one does not exclude the other. First, the design space is reduced with BCS before finishing with an optimizer. Fractions of the computational budget are reserved for each. After BCS depletes its function calls, the final Bayesian models may show relatively clear results, as in Figure 19, which is taken from a wing design example.

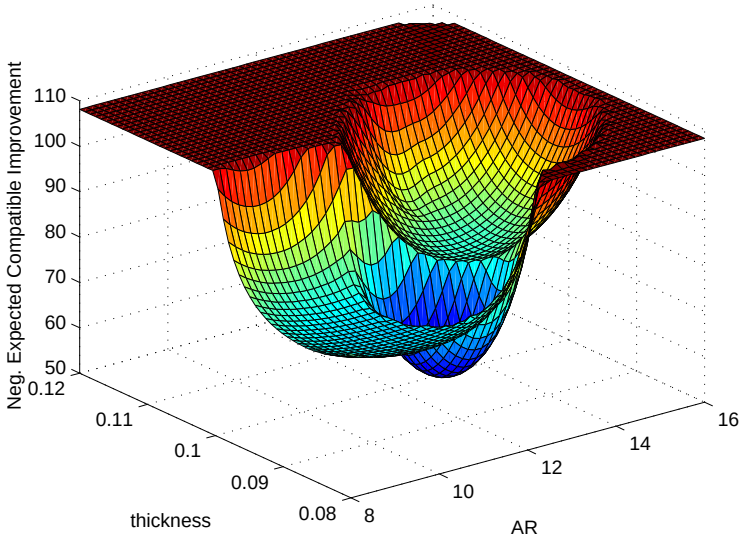

Figure 19: System sampling criterion after 5000 aerodynamic function calls

Here, the system level sampling criterion is shown with respect to two design variables; lower values correspond to higher expectation of improvement and probability of compatibility. Most of the design space is ruled out (red plateaus) based on the current best design. At this point, the design variable ranges may be constricted to the most promising region and the current best design can be used as an initial point for direct optimization.

The second option is to use the entire budget for BCS. The Bayesian models resulting from BCS can be used in ways similar to current DoE/surrogate modeling methods. They can give the designer qualitative information about favorable regions of the design space, such as objective and constraint sensitivities. In the end, the current best design is selected, and designer proceeds to the next stage, adding detailed design variables that had previously been screened out.

It is not necessary to choose one of these BCS strategies in advance. One great advantage of BCS is that it can be easily stopped, inspected, and restarted. For example, before BCS depletes its budget, it may be observed that Bayesian model variances are very low, the sample concentrates densely in one region, and there are

few exploratory excursions outside the region. This scenario tends to happen with simple function shapes with little numerical noise, such as algebraic test functions. The designer may become highly confident that BCS has learned enough and wish to stop prematurely and proceed to direct optimization. Alternately, he/she may simply take the current best point and proceed to detailed design.

Figure 20 summarizes the practical use of BCS in early design phases.

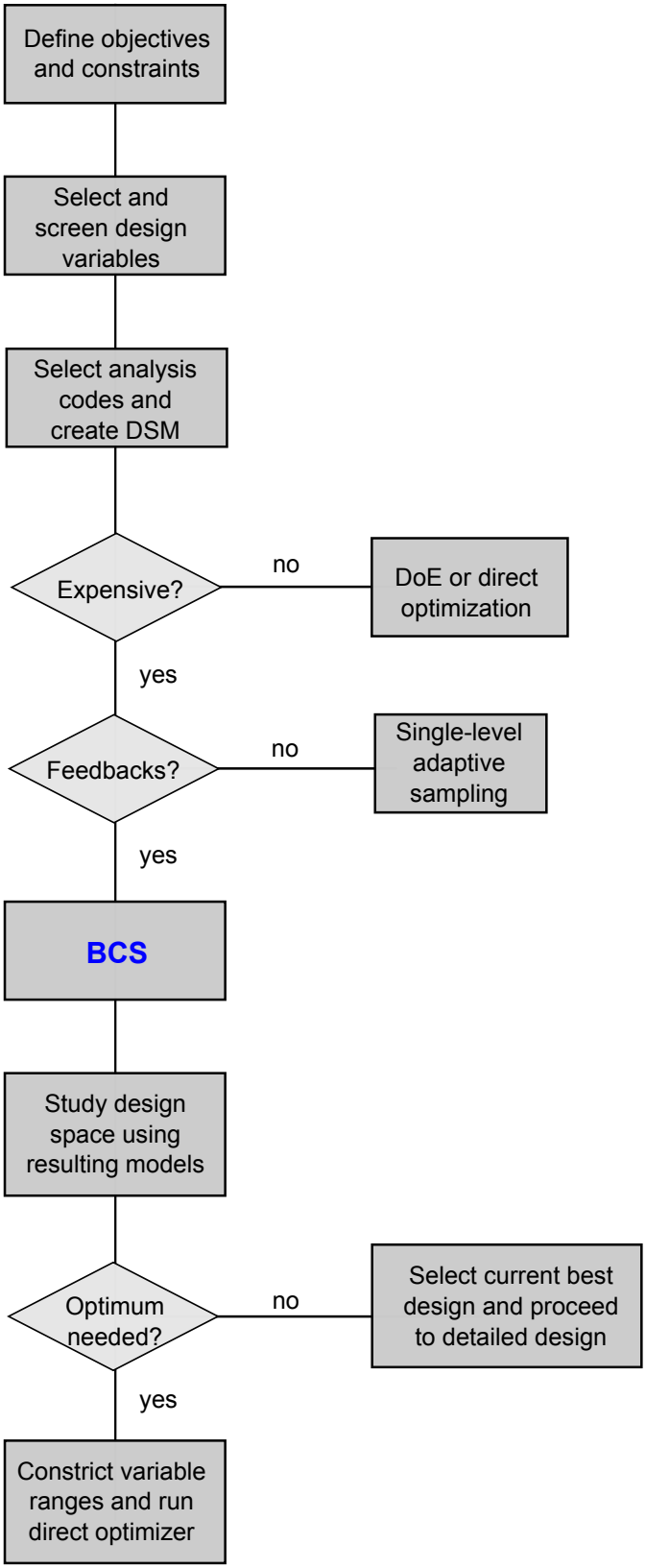

Figure 20: Conceptual/preliminary design method using BCS

CHAPTER III

BAYESIAN INFERENCE AND MODELING

In the previous chapter, Bayesian models were described only briefly before showing their use in the BCS architecture. It was assumed that a model could somehow be created and provide a posterior predictive distribution, which could then be used to compute useful quantities like expected improvement. This chapter gives a more detailed theoretical account of Bayesian inference and modeling. The end goal is to describe a sparse Bayesian model, which is based on a model type called a Relevance Vector Machine (RVM). These Bayesian models are also well-documented in machine learning and pattern recognition textbooks such as Bishop [19], and even a Matlab code for RVM is available from its original author under GNU public license [149]. Their theoretical development is not a contribution of this thesis, but their novel usage for adaptive sampling of design problems is. If a detailed mathematical derivation and theoretical justification is not required, then the reader may wish skip to the next chapter.

The general BCS architecture can notionally work with any generic Bayesian model, but computational expense is a major practical concern. In the sequential sampling scheme, the models are used to inexpensively select the most promising sample points for an expensive analysis. However, if the model training and evaluation are expensive compared to the analysis codes, there is no purpose to using BCS. Recall that although existing Kriging-based methods may be useful for modest single-level design problems, the modeling cost can become too high for larger scale, multi-level, multidisciplinary problems. Sparse Bayesian kernel machines have certain properties that make them efficient and well-suited for practical BCS.

The chapter covers the following:

1. Philosophy of Bayesian inference and modeling
2. Frequentist and Bayesian views of linear regression
3. Sparse Bayesian regression and the relevance vector machine (RVM)
4. Sampling criteria for adaptive learning

3.1 The Philosophy of Bayesian Inference and Models

The Bayesian paradigm has been explained in detail by authoritative sources. The purpose here is not to repeat their rigorous development of the subject, but to highlight the main philosophical reasons why Bayesian probability is appropriate for sequential sampling in design problems.

We can begin by defining a random variable. Strictly speaking, "random" is the opposite of "deterministic" [24]. However, in the Bayesian context, deterministic can mean *predictable* or *known* rather than having a unique outcome. So, *any variable that is unknown or unpredictable is a random variable.* Even if a variable is an output of a repeatable computer experiment, the observing subject (the experimenter) lacks knowledge of the precise output before running an analysis. This lack of knowledge makes the response variable random. The subject's belief in the value of a random variable can be expressed as a number: the subjective probability. The term subjective probability can be used interchangeably with Bayesian probability.

This notion of probability is often more intuitive than the frequentist perspective, where probability is a long-term relative frequency of an event in an asymptotically infinite series of observations. For example, if a person were to say, "I am 50% certain that Thomas Bayes was a minister," it is a Bayesian probability. Bayes's occupation is not a random event with replication error, where repeated observations will show

that he is a minister in half of all cases. Rather, the 50% probability indicates the speaker's equal uncertainty about the statement being true or false.

But what does it mean to say that there is a 70% probability that Thomas Bayes was a minister? The precise number seems to depend too much on the opinion of the observer, especially since the statement cannot be falsified by a large number of repeated experiments. In practice, it is the relative degree of knowledge that is more important than the particular number. If we know that one candidate aircraft design has a 70% probability of aero-structural compatibility and another design has an 80% probability, we will pick the latter and not worry about the precise meaning of "80%." Compared to 70%, it is closer to certain compatibility (100%) and farther from certain incompatibility (0%).

Despite this common-sense view of relative certainties, it is not an arbitrary choice to treat uncertainty with the formal mathematics of probability. For example, a famous 1946 work by Cox showed that a few, common-sense axioms about the numerical values of belief led to rules for the manipulation of these quantities that obey the rules of classical probability [34]. Other authors over the past seventy years have proposed different axioms for codifying degrees of belief, and they have also shown that these result in rules that are equivalent to probability [72]. Therefore, it is both intuitive and mathematically valid to think of degree of knowledge in terms of Bayesian probability.

This idea of probability is used in Bayesian inference and modeling. Inference is the estimation of parameters for a distribution after observing data from that distribution; modeling involves fitting a simplified expression to represent observed data. In this context, "Bayesian" methods are not sharply defined set but are rather a collection of related ideas and tools that tend to be used in inference and modeling. The first of these is subjective probability. Other common themes include:

- Bayes's theorem on posterior probability

- Marginalization (integrating out 'nuisance' parameters in probability distributions)
- Treatment of model parameters as random variables
- Improvement of a model by sequential observation and re-fitting
- "Ockham's razor" (a preference for models with low complexity)

Bayesian methods have become increasingly important in *supervised learning* tasks in the fields of machine learning and statistical pattern recognition. Supervised learning is essentially similar to surrogate modeling in engineering design: the learning algorithm is presented with a training data set $\mathbf{D}$ that consists of input variables $\mathbf{X}$ and outputs $\mathbf{T}$, using the notation from the previous chapter. Given a new input $\mathbf{x}^*$, the goal of supervised learning is to predict the corresponding response t^* using a predictive model based on observations $\mathbf{D}$. The predictive model has parameters $\mathbf{w}$ that are fit to the training data, and the challenge is to do this without overfitting the model to $\mathbf{D}$ so that the predictions generalize well to yet unseen data. With limited training data, however, there is uncertainty in the prediction. The frequentist paradigm tends to focus on uncertainty due to noise in the response t. However, the Bayesian paradigm focuses on the uncertainty of model parameters $\mathbf{w}$ or model form.

In a general case, the uncertainty in $\mathbf{w}$ is represented in two ways. First, there is a prior distribution $p(\mathbf{w})$ that represents our assumptions about $\mathbf{w}$ before observing any data. After the data $\mathbf{D}$ is observed, we can quantify how much of $\mathbf{D}$ is explained by different values of $\mathbf{w}$. This conditional probability is denoted $p(\mathbf{D}|\mathbf{w})$, and can be considered a function of $\mathbf{w}$. A distribution like $p(\mathbf{D}|\mathbf{w})$ that is interpreted as a functions of parameters $\mathbf{w}$ is conventionally called a *likelihood* function. The prior and likelihood can be used to find the posterior uncertainty on $\mathbf{w}$ through Bayes's theorem:

$$p(\mathbf{w}|\mathbf{D}) = \frac{p(\mathbf{D}|\mathbf{w})\, p(\mathbf{w})}{p(\mathbf{D})}$$

or

$$\texttt{posterior} = \frac{\texttt{likelihood} \times \texttt{prior}}{\textit{normalizing factor}}$$

Once the posterior $p(\mathbf{w}|\mathbf{D})$ is known, it can be used to find a posterior prediction $p(t^*|\, \mathbf{x}^*, \mathbf{D})$. The process of going from prior to likelihood to posterior to a posterior prediction is at the heart of Bayesian modeling. The following section steps through a linear regression example of the different levels of Bayesian treatment of $p(\mathbf{D}|\mathbf{w})$ and $p(t^*|\, \mathbf{x}^*, \mathbf{D})$. It first starts with a frequentist approach and progressively adds Bayesian elements, building up to the sparse Bayesian method.

3.2 From Frequentist to Bayesian Linear Regression

In Chapter 2, the frequentist approach to linear regression was explained in terms of minimizing an error function. The same procedure is now explained in terms of likelihood in order to build up an explanation of Bayesian modeling. The following discussion closely follows the notation and explanations in Duda, Hart, and Stork [47], Tipping and Smola [150], [149], Bishop [19], MacKay [95], and other common texts in pattern recognition and machine learning. In particular, the order of presentation is influenced by Tipping.

As before, we are given a data set

$$\mathbf{D} = \{\mathbf{X}, \mathbf{T}\}$$

where

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \vdots \\ \mathbf{x}^{(N)} \end{bmatrix}$$

and

$$\mathbf{T} = \begin{bmatrix} t\left(\mathbf{x}^{(1)}\right) \\ t\left(\mathbf{x}^{(2)}\right) \\ \vdots \\ t\left(\mathbf{x}^{(N)}\right) \end{bmatrix}.$$

A model form is also assumed:

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi} \approx t(\mathbf{x})$$

where the basis vector $\boldsymbol{\phi}$ is a column vector of L elements. A matrix $\boldsymbol{\Phi}$ is also defined such that each row corresponds to basis functions evaluated at an observed sample point.

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}(\mathbf{x}^{(1)})^T \\ \boldsymbol{\phi}(\mathbf{x}^{(2)})^T \\ \vdots \\ \boldsymbol{\phi}(\mathbf{x}^{(N)})^T \end{bmatrix}$$

Thus, if each basis vector $\boldsymbol{\phi}$ has L basis functions and there are N observed data points, then $\boldsymbol{\Phi}$ is $N \times L$.

Earlier, we fit the model by minimizing with respect to $\mathbf{w}$:

$$E(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} [y(\mathbf{x}^{(i)}, \mathbf{w}) - t^{(i)}]^2$$

This time, the curve fitting is explained in terms of maximizing a likelihood function $p(\mathbf{D}|\mathbf{w})$. Note that in this case, $p(\mathbf{D}|\mathbf{w})$ will be written as $p(\mathbf{T}|\mathbf{w})$ to be consistent with the cited literature. It is assumed that the inputs $\mathbf{X}$ that comprise part of $\mathbf{D}$ are fixed points (for example, from a DoE) rather than a stream of uncontrolled, observed data. $\mathbf{X}$ can optionally be dropped from the notation, and "observed data" will refer to $\mathbf{T}$.

3.2.1 Maximum Likelihood Estimation (MLE)

Maximum likelihood is a popular frequentist technique for model fitting. We assume that the model $y(\mathbf{x})$ correctly represents the underlying phenomenon $t(\mathbf{x})$ of which $\mathbf{T}$ is a sample, and any discrepancy from the model is random error with zero mean and unknown variance σ^2. For simplicity, this noise is assumed to be Gaussian, so that

$$t(\mathbf{x}) = y(\mathbf{x}, \mathbf{w}) + \epsilon$$

where $\epsilon \sim \mathcal{N}(0, \beta^{-1})$. Following the convention of machine learning literature, the inverse variance $\beta = (\sigma^2)^{-1}$ is called the *precision*. An alternate notation emphasizes that the uncertainty is over the response t,

$$p\left(t|\mathbf{x}, \mathbf{w}, \beta\right) = \mathcal{N}\left(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}\right) \tag{10}$$

If each sample point in the data set is assumed to be an independent observation from this distribution, then the probability of seeing all N elements of $\mathbf{T}$ is the product of the probabilities of each observation. This is expressed as

$$p\left(\mathbf{T}|\mathbf{X}, \mathbf{w}, \beta^{-1}\right) \sim \prod_{i=1}^{N} \mathcal{N}\left(t^{(i)}|y(\mathbf{x}^{(i)}, \mathbf{w}), \beta^{-1}\right) \tag{11}$$

In the present context, $p\left(\mathbf{T}|\mathbf{X}, \mathbf{w}, \beta^{-1}\right)$ is called the likelihood, similar to the more general $p(\mathbf{D}|\mathbf{w})$ mentioned earlier. For this distribution, we can assume a Gaussian with the general form

$$\mathcal{N}(z|\,\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(z-\mu)^2\right] \tag{12}$$

This expression is substituted into the likelihood function of Eq.(11). For later convenience, the logarithm of the likelihood function is used, yielding:

$$\ln p\left(\mathbf{T}|\mathbf{X}, \mathbf{w}, \beta\right) = -\frac{\beta}{2}\sum_{i=1}^{N}\left[y(\mathbf{x}^{(i)}, \mathbf{w}) - t^{(i)}\right]^2 + \frac{N}{2}\ln\beta - \frac{N}{2}\ln\left(2\pi\right) \tag{13}$$

To estimate the "true" value of $\mathbf{w}$, the maximum of (13) is found with respect to $\mathbf{w}$. The latter two terms do not depend on $\mathbf{w}$, and the coefficient β in the first term does not influence that stationary point. Without explicitly taking the derivative of the log-likelihood, we see that maximum likelihood solution $\mathbf{w}_{ML}$ is the same as the $\hat{\mathbf{w}}$ that minimized the sum-of-squares error function given earlier in (1):

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}[y(\mathbf{x}^{(i)}, \mathbf{w}) - t^{(i)}]^2$$

The other model parameter is the noise precision β. Taking the derivative of the log-likelihood with respect to β and setting it equal to zero, we find the maximum likelihood estimate β_{ML} is given by

$$\frac{1}{\beta_{ML}} = \frac{1}{N}\sum_{i=1}^{N}\left[y(\mathbf{x}^{(i)}, \mathbf{w}_{ML}) - t^{(i)}\right]^2 \tag{14}$$

At this stage, $\mathbf{w}_{ML}$ and β_{ML} can be taken as point estimates for a surrogate model

$$t(\mathbf{x}) \approx y(\mathbf{x}, \mathbf{w}_{ML}) = \mathbf{w}_{ML}^T\boldsymbol{\phi} \tag{15}$$

with error variance β_{ML}^{-1}. However, taking a slightly Bayesian point of view, we can intepret this as a predictive distribution

$$p\left(t^*|\,\mathbf{x}^*, \mathbf{w}_{ML}, \beta_{ML}\right) \sim \mathcal{N}\left(t^*|\,y(\mathbf{x}^*, \mathbf{w}_{ML}), \beta_{ML}^{-1}\right). \tag{16}$$

Next, we introduce some Bayesian elements to this method as an intermediate step.

3.2.2 Maximum *a Posteriori* (MAP) Estimation

The maximum *a posteriori* method introduces a prior distribution on the model parameters $\mathbf{w}$ and hyperparameters. It is still not a full Bayesian treatment, but it is an intermediate step that illustrates many of the important principles.

With MLE, the distribution over t depends on a point estimate of $\mathbf{w}$. In the Bayesian setting, $\mathbf{w}$ is uncertain, depending on a distribution over $\mathbf{w}$ with precision

α. For this example, it is assumed to be Gaussian:

$$\begin{aligned} p(\mathbf{w}|\,\alpha) &= N\left(\mathbf{w}|\,\mathbf{0}, \alpha^{-1}\,\mathbf{I}\right) \\ &= \left(\frac{\alpha}{2\pi}\right)^{M/2} \exp\left[-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}\right] \end{aligned} \tag{17}$$

where M is the number of basis functions $\boldsymbol{\phi}$ or elements of $\mathbf{w}$. For now, α is assumed to be known.

Earlier, a single value of $\mathbf{w}$ influenced a distribution on $\mathbf{T}$. In this case, a single value of α influences a distribution on $\mathbf{w}$, which in turn influences a distribution on $\mathbf{T}$. This can be used to find a posterior distribution on $\mathbf{w}$ through Bayes's theorem:

$$\begin{aligned} \texttt{posterior} &= \frac{\texttt{likelihood} \times \texttt{prior}}{\textit{normalizing factor}} \\ p\left(\mathbf{w}|\,\mathbf{T}, \alpha, \beta\right) &= \frac{p\left(\mathbf{T}|\,\mathbf{w}, \beta\right)\; p\left(\mathbf{w}|\,\alpha\right)}{p(\mathbf{T}|\,\alpha, \beta)} \end{aligned} \tag{18}$$

Thus, the posterior distribution over $\mathbf{w}$ depends on the likelihood function (similar to the previous result from MLE) as well as a prior belief about $\mathbf{w}$ given some parameter α. A parameter like α that influences the distribution of another model parameter is called a *hyperparameter*.

Since we have assumed a Gaussian prior and likelihood, the posterior on $\mathbf{w}$ is also Gaussian, given by

$$p\left(\mathbf{w}|\,\mathbf{T}, \alpha, \beta\right) \sim \mathcal{N}(\mathbf{w}|\,\mathbf{m}, \mathbf{S}) \tag{19}$$

where

$$\mathbf{m} = \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \frac{\alpha}{\beta}\mathbf{I}\right)^{-1} \boldsymbol{\phi}^T\mathbf{T} \tag{20}$$

$$\mathbf{S} = \frac{1}{\beta}\left(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \frac{\alpha}{\beta}\mathbf{I}\right)^{-1} \tag{21}$$

Earlier, MLE found a point estimate of a single $\mathbf{w}$, but now there is a whole distribution of possible values. A full Bayesian account would integrate over this distribution to take a weighted sum of all possible values of $\mathbf{w}$. But for now, we will show a partially Bayesian 'shortcut' method.

The *maximum a posteriori* (MAP) method finds the most probable $\mathbf{w}$ at the peak of the posterior distribution of (18). Since the denominator has no dependence on $\mathbf{w}$, the MAP method amounts to maximizing the numerator with respect to $\mathbf{w}$. Once again, it is more convenient to maximizing the logarithm of the probabilities rather than the probabilities themselves. Substituting (13) and (17) into (18), it is found that maximizing the log-posterior is equivalent to minimizing an error function

$$E_{MAP}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}\left[y(\mathbf{x}^{(i)}, \mathbf{w}) - t^{(i)}\right]^2 + \frac{\alpha}{2\beta}\mathbf{w}^T\mathbf{w}. \tag{22}$$

The solution for minimum E_{MAP} is denoted $\mathbf{w}_{MAP}$. Note that E_{MAP} is the sum-of-squares error with an added term that penalizes non-zero values in $\mathbf{w}$. The coefficient $\frac{\alpha}{2\beta}$ acts as a penalty weight (or *regularization* parameter). Note that when the prior precision α is high, then curve fitting tends to favor models with elements of $\mathbf{w}$ close to zero—in other words, it favors turning basis functions 'off' and prefers less complex models. This foreshadows a mechanism of parsimony in sparse Bayesian models that will be described later.

3.2.3 Marginalization and the Posterior Predictive Distribution

The MAP technique has introduced some Bayesian elements such as a likelihood and prior distribution on $\mathbf{w}$ as well as Bayes's theorem and the hyperparameter α. However, it has not reached the goal of a predictive distribution on the response $p(t^*|\,\mathbf{x}^*, \mathbf{T})$ for a new point $\mathbf{x}^*$. To do this, we must reach into the Bayesian toolbox for *marginalization* — the technique of integrating out parameters that are not of main concern. Marginalization requires the sum and product rules of probability.

$$p(X) = \sum_{Y} p(X, Y) \qquad \text{(Sum Rule)}$$

$$p(X, Y) = p(X|Y)\, p(Y) \qquad \text{(Product Rule)}$$

Applying these rules to (10) and (19) yields

$$p\left(t^*\middle|\,\mathbf{x}^*,\mathbf{T},\alpha,\beta\right)=\int p\left(t^*\middle|\,\mathbf{x}^*,\mathbf{w}\right)\;p\left(\mathbf{w}\middle|\,\mathbf{T},\alpha,\beta\right)\mathrm{d}\mathbf{w}.$$

The next section will explain the estimation of α and β. For now, they are still assumed to be known constants, so they are temporarily dropped from the notation, following the convention of Bishop [19]:

$$p\left(t^*\middle|\,\mathbf{x}^*,\mathbf{T}\right)=\int p\left(t^*\middle|\,\mathbf{x}^*,\mathbf{w}\right)p(\mathbf{w}\middle|\,\mathbf{T})\,\mathrm{d}\mathbf{w} \tag{23}$$

The result of marginalization is that parameters $\mathbf{w}$ do not appear in the posterior predictive distribution on the left side. In the current example, (23) contains two Gaussians in the integrand, so the solution is also Gaussian:

$$p\left(t^*\middle|\,\mathbf{x}^*,\mathbf{T}\right)=\mathcal{N}\left(t^*|\mu(\mathbf{x}^*),\sigma^2(\mathbf{x}^*)\right) \tag{24}$$

where mean $\mu(\mathbf{x}^*)$ and variance $\sigma^2(\mathbf{x}^*)$ are

$$\mu(\mathbf{x}^*)=\beta\,\boldsymbol{\phi}(\mathbf{x}^*)^T\mathbf{H}\sum_{i=1}^{N}\left[\boldsymbol{\phi}(\mathbf{x}^{(i)})\,t^{(i)}\right] \tag{25}$$

$$\sigma^2(\mathbf{x}^*)=\beta^{-1}+\boldsymbol{\phi}(\mathbf{x}^*)^T\mathbf{H}\,\boldsymbol{\phi}(\mathbf{x}^*) \tag{26}$$

and where

$$\mathbf{H}=\left[\alpha\,\mathbf{I}+\beta\;\sum_{i=1}^{N}\boldsymbol{\phi}(\mathbf{x}^{(i)})\;\boldsymbol{\phi}(\mathbf{x}^{(i)})^T\right]^{-1}. \tag{27}$$

In (26), the β^{-1} term represents variance due to noise present in the data while the second term represents uncertainty due to the model.

3.2.4 Full Bayesian Treatment

In the previous section, the posterior predictive distribution had an analytic solution. However, it was still not a fully Bayesian account, because the hyperparameters α and β were assumed constants. Also, for greater generality, we wish to use a vector

of hyperparameters $\boldsymbol{\alpha}$. There is an α_i and independent Gaussian prior $p(w_i|\,\alpha_i)$ associated with each weight w_i. The total weight prior is given by

$$p(\boldsymbol{w}|\,\boldsymbol{\alpha}) = \prod_{i=1}^{M} \mathcal{N}\left(w_i|\,0, \alpha_i^{-1}\right). \tag{28}$$

To compute a fully Bayesian posterior prediction, we could mirror the steps of the previous section, create hyperpriors over α and β, and marginalize over parameters and hyperparameters to give a full posterior prediction for t^*.

$$p\left(t^*|\,\mathbf{x}^*, \mathbf{X}, \mathbf{T}\right) = \iiint p\left(t^*|\,\mathbf{x}^*, \mathbf{w}, \beta\right)\; p\left(\mathbf{w}, \boldsymbol{\alpha}, \beta|\,\mathbf{T}\right) \mathrm{d}\mathbf{w}\; \mathrm{d}\boldsymbol{\alpha}\; \mathrm{d}\beta \tag{29}$$

This requires a posterior over all model parameters and hyperparameters, which is the second term in the integrand. By Bayes's theorem, this is given by

$$p\left(\mathbf{w}, \boldsymbol{\alpha}, \beta|\,\mathbf{T}\right) = \frac{p\left(\mathbf{T}|\,\mathbf{w}, \beta\right)\; p\left(\mathbf{w}|\,\boldsymbol{\alpha}\right)\; p\left(\boldsymbol{\alpha}\right)\; p\left(\beta\right)}{p(\mathbf{T})} \tag{30}$$

where the denominator is found by marginalizing the probability of the data over all parameters and hyperparameters:

$$p\left(\mathbf{T}\right) = \iiint p\left(\mathbf{T}|\,\mathbf{w}, \beta\right)\; p\left(\mathbf{w}|\boldsymbol{\alpha}\right)\; p\left(\boldsymbol{\alpha}\right)\; p\left(\beta\right)\; \mathrm{d}\mathbf{w}\; \mathrm{d}\boldsymbol{\alpha}\; \mathrm{d}\beta \tag{31}$$

Unfortunately, (31) is usually analytically intractable. Recent research in machine learning has focused on approximate techniques, including the Laplace method, variational methods, and Type II maximum likelihood estimation [19]. The last method is used in this study because of its simplicity.

3.2.5 Type-II Maximum Likelihood (Evidence Approximation)

Since (31) is intractable, the full posterior over parameters in (30) is factorized:

$$p\left(\mathbf{w}, \boldsymbol{\alpha}, \beta|\,\mathbf{T}\right) = p\left(\mathbf{w}|\,\mathbf{T}, \boldsymbol{\alpha}, \beta\right)\; p\left(\boldsymbol{\alpha}, \beta|\,\mathbf{T}\right) \tag{32}$$

Substituting this result into Eq.(29), the posterior predictive distribution can be rewritten as

$$p\left(t^*|\,\mathbf{x}^*, \mathbf{T}\right) = \iiint p\left(t^*|\,\mathbf{x}^*, \mathbf{w}, \beta\right)\; p\left(\mathbf{w}|\,\mathbf{T}, \boldsymbol{\alpha}, \beta\right)\; p\left(\boldsymbol{\alpha}, \beta|\,\mathbf{T}\right) \mathrm{d}\mathbf{w}\; \mathrm{d}\boldsymbol{\alpha}\; \mathrm{d}\beta. \tag{33}$$

The first term in the integrand is (10). The second term is similar to the weight posterior (19) from earlier, except for some modification because of the vector $\boldsymbol{\alpha}$:

$$p\left(\mathbf{w}|\ \mathbf{T}, \boldsymbol{\alpha}, \beta\right) = \mathcal{N}(\mathbf{w}|\ \mathbf{m}, \mathbf{S}) \tag{34}$$

where

$$\mathbf{m} = \beta \mathbf{S} \boldsymbol{\Phi}^T \mathbf{T} \tag{35}$$

$$\mathbf{S} = \left(\mathbf{A} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}\right)^{-1} \tag{36}$$

$$\mathbf{A} = \mathrm{diag}(\alpha_i) \tag{37}$$

However, the third term in the integrand of (33) is difficult to compute and must be approximated. The hyperparameter posterior $p\left(\boldsymbol{\alpha}, \beta|\ \mathbf{T}\right)$ is replaced by a δ-function at its most probable values, $\boldsymbol{\alpha}_{MP}$ and β_{MP}. These most probable values can be approximated more easily if the hyperparameter posterior is expanded with Bayes's theorem.

$$p(\boldsymbol{\alpha}, \beta|\ \mathbf{T}) = \frac{p(\mathbf{T}|\ \boldsymbol{\alpha}, \beta)\, p(\boldsymbol{\alpha})\, p(\beta)}{p(\mathbf{T})}. \tag{38}$$

Eq.(38) must be maximized with respect to $\boldsymbol{\alpha}$ and β. The denominator has no $\boldsymbol{\alpha}$ or β dependence and is ignored. The hyperpriors $p(\boldsymbol{\alpha})$ and $p(\beta)$ can be set to an arbitrary form. If they are set to flat, uninformative distributions, then only the first term in the numerator must be maximized. This term, $p(\mathbf{T}|\ \boldsymbol{\alpha}, \beta)$ is called the 'marginal likelihood,' since it is a likelihood function in terms of hyperparameters where the weights $\mathbf{w}$ have been integrated out. The marginal likelihood is also called the 'hyperparameter evidence' in machine learning literature, such as MacKay [95].

In the special case of log-uniform hyperpriors, the marginal likelihood can be

expressed analytically:

$$p(\mathbf{T}|\,\boldsymbol{\alpha},\beta) = \int p(\mathbf{T}|\,\mathbf{w},\beta)\,p(\mathbf{w}|\,\boldsymbol{\alpha})\mathrm{d}\mathbf{w} \tag{39}$$

$$= (2\pi)^{-\frac{N}{2}}\left|\beta^{-1}\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^{T}\right|^{-\frac{1}{2}}\exp\left[-\frac{1}{2}\mathbf{T}^{T}\left(\beta^{-1}\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^{T}\right)^{-1}\mathbf{T}\right]. \tag{40}$$

This is a Gaussian and can be rewritten as

$$p(\mathbf{T}|\,\boldsymbol{\alpha},\beta) \sim \mathcal{N}(\mathbf{T}|\,\mathbf{0},\mathbf{C}) \tag{41}$$

where

$$\mathbf{C} = \beta^{-1}\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^{T}. \tag{42}$$

Eq.(40) can be numerically optimized with respect to hyperparameters to yield the most probable values, $\boldsymbol{\alpha}_{MP}$ and β_{MP}. The process is called "Type-II Maximum Likelihood" due to its similarity with the earlier MLE method, which optimized a likelihood function with respect to $\mathbf{w}$. Previous research in machine learning has described several methods for optimization, and they will not be repeated here [19], [149]. Once the most probable values are found, the marginal likelihood (evidence) can be approximated by δ-functions at its mode:

$$p(\mathbf{T}|\,\boldsymbol{\alpha},\beta) \approx \delta(\boldsymbol{\alpha}_{MP},\beta_{MP}). \tag{43}$$

This procedure is also known as "evidence approximation" [95].

At this point, we have analytic solutions or estimates for all of the terms in the full posterior predictive distribution (33). Substituting terms, we get:

$$p\left(t^{*}|\,\mathbf{x}^{*},\mathbf{X},\mathbf{T}\right) \approx \iiint p\left(t^{*}|\,\mathbf{x}^{*},\mathbf{w},\beta\right)\,p\left(\mathbf{w}|\,\mathbf{T},\boldsymbol{\alpha},\beta\right)\,\delta(\boldsymbol{\alpha}_{MP},\beta_{MP})\mathrm{d}\mathbf{w}\,\mathrm{d}\boldsymbol{\alpha}\,\mathrm{d}\beta$$

$$= \int p\left(t^{*}|\,\mathbf{x}^{*},\mathbf{w},\beta_{MP}\right)\,p\left(\mathbf{w}|\,\mathbf{T},\boldsymbol{\alpha}_{MP},\beta_{MP}\right)\,\mathrm{d}\mathbf{w}. \tag{44}$$

The integrand again contains two Gaussians, so the solution is Gaussian:

$$p\left(t^{*}|\,\mathbf{x}^{*},\mathbf{T}\right) \sim \mathcal{N}\left(t^{*}(\mathbf{x}^{*})|\,y(\mathbf{x}^{*}),\sigma^{2}(\mathbf{x}^{*})\right) \tag{45}$$

where

$$y(\mathbf{x}^*) = \mathbf{m}^T \boldsymbol{\phi}(\mathbf{x}^*) \tag{46}$$

$$\sigma^2(\mathbf{x}^*) = \beta_{MP}^{-1} + \boldsymbol{\phi}\,(\mathbf{x}^*)^T \mathbf{S} \boldsymbol{\phi}(\mathbf{x}^*) \tag{47}$$

and

$$\mathbf{S} = \left(\beta_{MP} \boldsymbol{\Phi}^T \boldsymbol{\Phi} + \mathbf{A}\right)^{-1} \tag{48}$$

$$\mathbf{m} = \beta_{MP}\, \mathbf{S}\, \boldsymbol{\Phi}^T \mathbf{T} \tag{49}$$

$$\mathbf{A} = \text{diag}(\alpha_{_{MP,i}}) \tag{50}$$

In the predictive variance of Eq.(47), the β_{MP}^{-1} term can be considered a global uncertainty that cannot be explained by any of the basis functions, and the $\boldsymbol{\phi}\,(\mathbf{x}^*)^T \mathbf{S} \boldsymbol{\phi}(\mathbf{x}^*)$ represents a local uncertainty associated with the basis functions and observed data.

3.3 Sparse Bayesian Modeling

The evidence approximation procedure maximizes the evidence with respect to hyper-parameters α_i. However, this optimization often causes many α_i to tend to infinity. Recall that prior for an individual weight parameter is

$$p(w_i|\, \alpha_i) = \mathcal{N}\left(w_i|\, 0, \alpha_i^{-1}\right) \tag{51}$$

and the posterior over w_i is

$$p(w_i|\, \mathbf{T}, \alpha_i, \beta) \propto p(\mathbf{T}|\, w_i, \beta)\, p(w_i|\, \alpha_i). \tag{52}$$

As α_i becomes large, the probability mass of this posterior concentrates around $w_i = 0$. For practical Bayesian modeling, w_i is simply set equal to zero if α_i exceeds some threshold (like 10^8). This has the effect of pruning the basis function ϕ_i, which makes the model *sparse*.

There are many possible basis functions and algorithms for the Bayesian evidence procedure. Many of them use operations on matrices whose sizes depend on the

number of basis functions M, the sample size N, or both. Kriging and many kernel-based models have parameters or bases fixed to sample points, so $M = N$. As can be seen by expressions like (48), the modeling procedure may require factorization or even inversion of such M-dimensional matrices. Since the computational expense typically scales between $O(M^2)$ and $O(M^3)$, there is a major advantage in pruning unnecessary basis functions.

The mechanism behind this pruning is called *automatic relevance determination* (ARD). During the evidence procedure, each candidate basis function is implicitly compared with isotropic, zero-mean noise. If the basis does not improve the evidence more than this simple noise, it is rejected and the global noise level (represented by β) is adjusted by an appropriate amount. The sparse Bayesian models therefore have a built-in complexity control that avoids overfitting and an excessive number of parameters. This may be considered a form of "Ockham's razor," or the principle of parsimonious explanation.

A particular sparse Bayesian model called a relevance vector machine (RVM) is first described to motivate a more detailed explanation of ARD and sparsity.

3.3.1 The Relevance Vector Machine

The relevance vector machine (RVM) is a Bayesian model that uses the evidence procedure of the previous section to train a sparse set of basis functions [150]. RVM bases are kernels functions $k(\mathbf{x}, \mathbf{x}^{(i)})$:

$$y(\mathbf{x}) = \sum_{i=1}^{N} w_i k(\mathbf{x}, \mathbf{x}^{(i)}) + w_0. \tag{53}$$

Although many kernel types can be used, the present study uses kernels of the form

$$k(\mathbf{x}, \mathbf{x}') = \exp\left[-\left(\frac{\|\mathbf{x} - \mathbf{x}'\|}{c}\right)^p\right]. \tag{54}$$

The case where p = 2 is a Gaussian kernel and p = 1 is a Laplace kernel, named for their similarity to the respective distributions. In this sparse Bayesian model, the kernels that remain after pruning are called "relevance vectors."

Note that (54) with p = 2 has nothing to do with a Gaussian probability density function despite its form and name. It is only a bump-shaped basis function that expresses the influence or similarity between two points. Typically, these functions are fixed to sample points. Second, even though the model form of (53) and (54) is identical to a Gaussian radial basis function, it is different because Bayesian evidence procedure is used to find a predictive distribution.

The c term is a characteristic width of the kernel function. In intuitive terms, it is a length scale at which the response at a sample point influences the response at a new point. It is the only 'nuisance' parameter that must be tuned outside of the evidence procedure, so it is set by direct minimization of the RMS error of the predictive mean, computed by k-fold cross-validation.

The main advantage of the RVM is its speed compared to Kriging. The following three figures show an example using the 'sinc' function corrupted with synthetic Gaussian noise: $y = \sin(x)/x + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.05)$. The figures show the true function and 100 data points, and then the predictive distributions with Kriging and RVM, respectively.

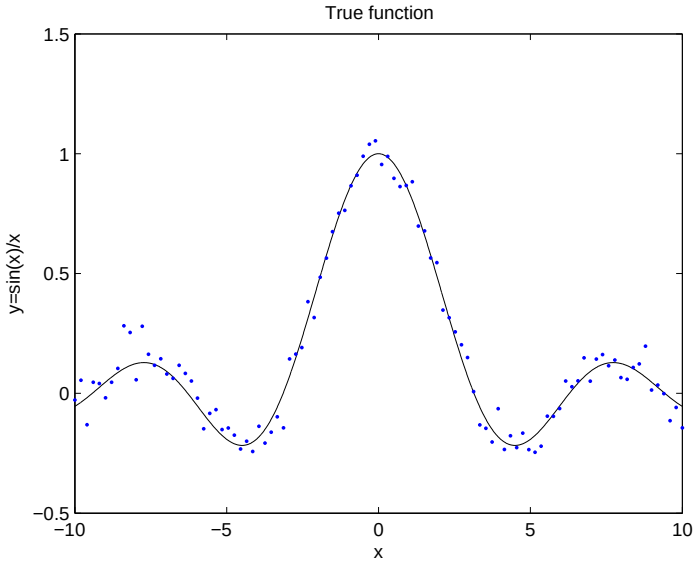

Figure 21: $\sin(x)/x + \epsilon$

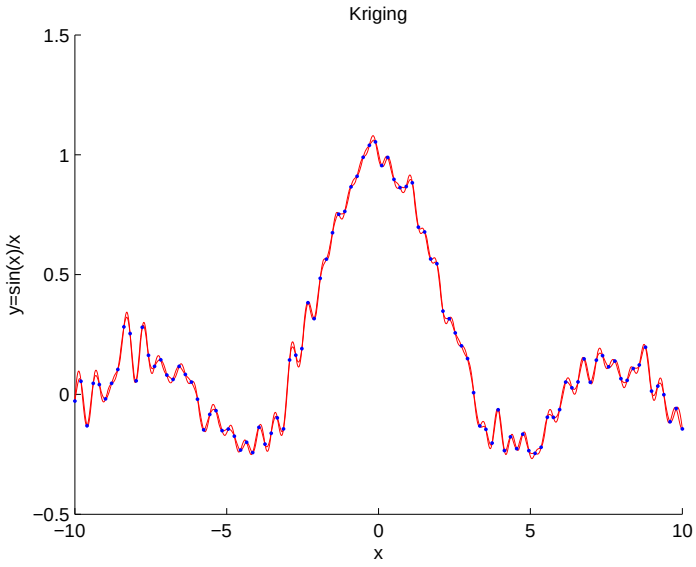

Figure 22: Kriging predictive distribution, $\pm\sigma$

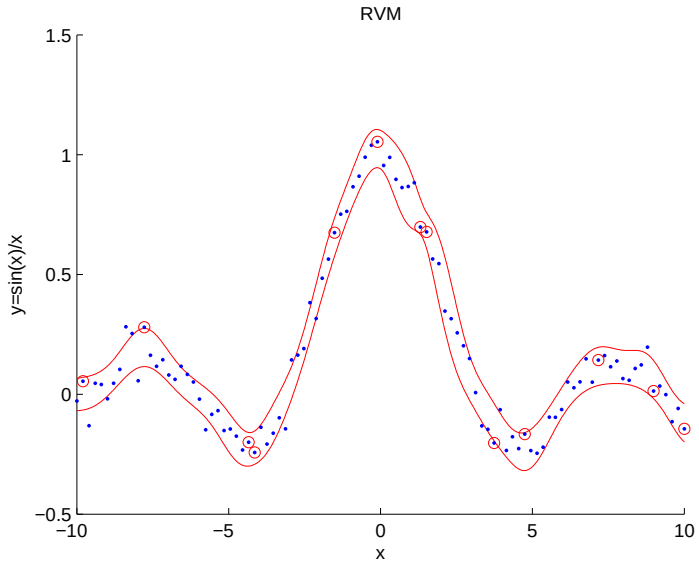

Figure 23: RVM predictive distribution, $\pm\sigma$

In Figure 22 and 23, the curves show contours of $\sigma = 1$. In the Kriging example, the predictive variances are so small that it may be difficult to distinguish upper and lower $\pm\sigma$ contours. In figure 23, the circled points represent the relevance vectors—kernel functions at other points are inactive.

Table 1 shows the model training times for various sample sizes. The models were fit using Matlab 2009a on an Intel i7 with 2.93 GHz, 12GB RAM.

Table 1: Model training times

Sample size	Kriging	RVM
10	0.24 s	0.30 s
500	180	0.26
2000	2200	0.0000097
5000	14000	4.2

Even with a single design variable, Kriging requires over three hours to fit a sample of 5000. For practical design problems of $O(10)$ design variables and sample sizes of $O(10^3)$, the computational cost of Kriging can be comparable to the cost of high fidelity analysis and may not even be possible on a typical desktop PC. Also recall that

the BCS method requires models for the objective function and every compatibility condition. These models must be updated at every iteration. A comparison of the other merits of Kriging and RVM in BCS is almost moot in some cases, since Kriging can be impractically expensive.

Nonetheless, there are some differences in the models' predictions that have consequences for adaptive sampling. In Figure 22, it can be seen that Kriging is an interpolator — the model passes through every observed point, and the predictive variance shrinks to zero at each point. On the other hand, RVM is a regression, and the predictive variance merely constricts in regions of high certainty. In 'noisy' regions[1], the automatic relevance determination (ARD) mechanism causes variance to increase rather than using complicated features in the model to achieve a close fit. In informal terms: "If the sample data show 'squiggles' that look a lot like patternless noise, then simply call them noise and don't try too hard to explain every local feature." ARD is described in the following section.

Two main consequences result from these differences between Kriging and RVM. First, Kriging tends to overfit a 'noisy' function. Even small scale features from noisy data are resolved, so a Kriging-based adaptive learning algorithm can easily become trapped in false local minima. Such functions are common in applications like CFD, where gridding or roundoff error can lead to non-smoothness unrelated to the governing physics. The RVM-based method is relatively insensitive to noisy functions. Second, the predictive variance of RVM does not monotonically decrease as sample density increases in a region. Because of this, a RVM-based adaptive sampling algorithm usually cannot act as an optimizer. As long as the RVM predictive variance detects isotropic noise ($1/\beta_{MP}$), there will remain some irreducible uncertainty even in very densely sampled region containing an optimum.

[1]In the context of deterministic computer experiments, "noise" loosely refers to non-smooth, multimodal features due to numerical error rather than replication error.

In summary, the tradeoff for not converging on false optima due to noise is that the adaptive sampling will not converge at all. The lack of convergence is not necessarily a problem for an adaptive sampling scheme, for it will simply continue its exploration/exploitation of the design space. A corollary to this discussion is that if the analysis functions are very smooth (e.g. algebraic functions), then an RVM-based adaptive sampler can act as an optimizer. This behavior is demonstrated in the next chapter.

As a side note, Table 1 shows a curious phenomenon where the speed of RVM does not vary monotonically with sample size. There are two trends that affect training speed. First, as the sample size increases, it is easier for RVM to ascribe 'noise-like' features to the global uncertainty level represented by β_{MP}^{-1}, prune excess basis functions earlier, and quickly fit the remaining bases. However, as the sample size increases, the cost of initially large matrix operations overtakes the savings from early pruning.

3.3.2 Automatic Relevance Determination

After introducing a particular type of sparse Bayesian model, a general principle called automatic relevance determination can be explained. This is the mechanism by which basis functions are pruned to produce a sparse model.

The main step of the evidence procedure is to maximize the marginal likelihood. That is, the hyperparameters $\boldsymbol{\alpha}$ and β are adjusted so to maximize the probability of the vector of observed responses $\mathbf{T}$. Again, the marginal likelihood is given by (41) and (42) as:

$$p(\mathbf{T}|\,\boldsymbol{\alpha}, \beta) \sim \mathcal{N}(\mathbf{T}|\,\mathbf{0}, \mathbf{C})$$

where

$$\mathbf{C} = \beta^{-1}\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^{T}.$$

The covariance matrix $\mathbf{C}$ has two parts. The first term depends only on β and is an isotropic variance that represents equal uncertainty in all directions in $\mathbf{T}$-space. The second term represents uncertainty in the model prediction, given $\boldsymbol{\alpha}$ settings.

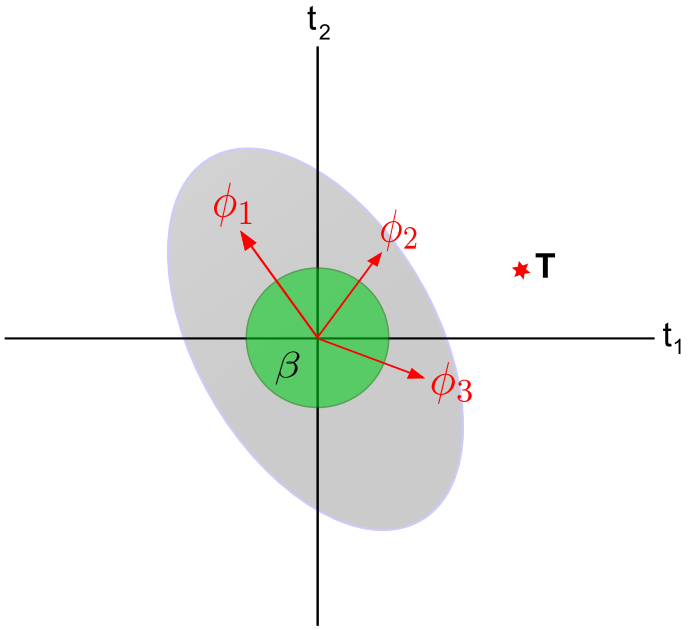

Figure 24: Mechanism for sparsity

Figure 24 shows a plot in $\mathbb{R}^N$, where each of N dimensions corresponds to each sample response. For simplicity, the example shows a sample of $N = 2$ designs. The vector $\mathbf{T}$ shown by the star represents the responses for these two designs. The ellipse represents a contour of unit Mahalanobis distance[2] of the marginal likelihood $p(\mathbf{T}|\,\boldsymbol{\alpha}, \beta) \sim \mathcal{N}(\mathbf{T}|\,\mathbf{0}, \mathbf{C})$. The goal of the evidence maximization is to adjust this distribution by manipulating $\boldsymbol{\alpha}, \beta$ until the most probability mass lies near the observed responses $\mathbf{T}$.

The arrows represent the directions and weighted effects of basis functions $\boldsymbol{\phi}$

[2] The Mahalanobis distance is a distance from the mean of a distribution normalized in terms of the covariance matrix. For a vector $\mathbf{X}$ and a distribution with mean $\boldsymbol{\mu}$ and covariance $\mathbf{C}$, the Mahalanobis distance is given by:

$$D_M = \sqrt{(\mathbf{X} - \boldsymbol{\mu})\mathbf{C}^{-1}(\mathbf{X} - \boldsymbol{\mu})}.$$

For the simple case with no covariance (if the covariance matrix has only diagonal terms), the Mahalanobis distance is Euclidean distance where length in each dimension is normalized by its standard deviation.

given the sample inputs. They indirectly affect the marginal likelihood through the $\mathbf{\Phi A}^{-1}\mathbf{\Phi}^T$ term of $\mathbf{C}$, and the strength of each ϕ_i's influence is controlled by its hyperparameter α_i. As the value of α_i approaches zero, the ellipse tends to stretch in the direction of ϕ_i. If α_i becomes very large, ϕ_i has little effect on the ellipse.

Part of the ellipse shape is due to an inner circle (darkened). This circle represents the contribution to $\mathbf{C}$ from its first term $\beta^{-1}\mathbf{I}$. Again, this is the directionless uncertainty that cannot be explained by any basis function. During the evidence maximization procedure, the effect of adjusting each α_i is compared to the effect of increasing β. That is, the effect of stretching the ellipse along the direction of ϕ_i is compared to the effect of increasing its circular component. If adjusting the α_i cannot make the data set $\mathbf{T}$ more likely than simply increasing the circle, then the i-th basis function is dropped and the circle radius is adjusted by a new β. In the figure, ϕ_1 is not closely aligned to the data set $\mathbf{T}$ as are ϕ_2 and ϕ_3. Stretching the ellipse in the ϕ_1 direction will not make $\mathbf{T}$ more likely than inflating the circular component, so the evidence procedure will probably increase α_1 beyond some threshold value and the the basis function will likely be dropped. This mechanism is called automatic relevance determination, and it can be implemented along with the evidence procedure by enforcing a cutoff theshold on high values of $\boldsymbol{\alpha}$.

Sparse Bayesian methods and RVMs use ARD to achieve faster speeds than other methods like Kriging. This mechanism is a type of Ockham's razor—or in a stronger form, the principle of parsimony. The principle of parsimony is a heuristic rule of preferring the simplest explanation for an observation. In the present case, the basis functions are candidate explanations for the data. If a basis cannot explain data better than random noise, then using that basis in a model could result in unnecessary complexity and possible overfitting, so it is therefore pruned.

3.3.2.1 Consequences of Sparsity

Consider again the predictive distributions on the response t that results from such a Bayesian model. Repeating Equations 45 through 47 from earlier, the prediction for response t^* at a new point $\mathbf{x}^*$ is given by:

$$p\left(t^* | \, \mathbf{x}^*, \mathbf{T}\right) \sim \mathcal{N}\left(t^*(\mathbf{x}^*) | \, y(\mathbf{x}^*), \sigma^2(\mathbf{x}^*)\right) \tag{55}$$

where

$$y(\mathbf{x}^*) = \mathbf{m}^T \boldsymbol{\phi}(\mathbf{x}^*) \tag{56}$$

$$\sigma^2(\mathbf{x}^*) = \beta_{MP}^{-1} + \boldsymbol{\phi}\left(\mathbf{x}^*\right)^T \mathbf{S} \boldsymbol{\phi}(\mathbf{x}^*) \tag{57}$$

In the variance in 57, the β_{MP} is associated with the isotropic uncertainty shown as a circle in Figure 24 and the latter term is due to the basis functions that remain after pruning. Figure 25 shows a more intuitive view of the these two elements in terms of the predictive variance of input x and output t. The figure shows 50 sample points of the test function $t = \sin(x)/x + \epsilon$, where the uniform random distribution $\epsilon \sim \mathcal{U}(0, 0.2)$ is used to make the function less smooth and more like a typical, "noisy" engineering analysis code. The solid red lines show the mean prediction with $\pm$ 1 predictive standard deviation. The dashed blue lines show only the contribution to this standard deviation of the β_{MP} term. It can be seen that this isotropic term associated with the circle in the earlier Figure 24 appears as a constant, global component of predictive uncertainty in the dashed blue lines. The consequence of this constant uncertainty associated with β_{MP} is that the sparse Bayesian model is not an interpolator like simple Kriging, but rather a regression method. The predictive variance does not constrict to zero at observed points, but has a global, minimum width. The consequence of this is that adaptive sampling based on a sparse Bayesian model often cannot converge as an optimizer. Another consequence is that unfavorable regions are not easily eliminated after a few sample points, because some global uncertainty

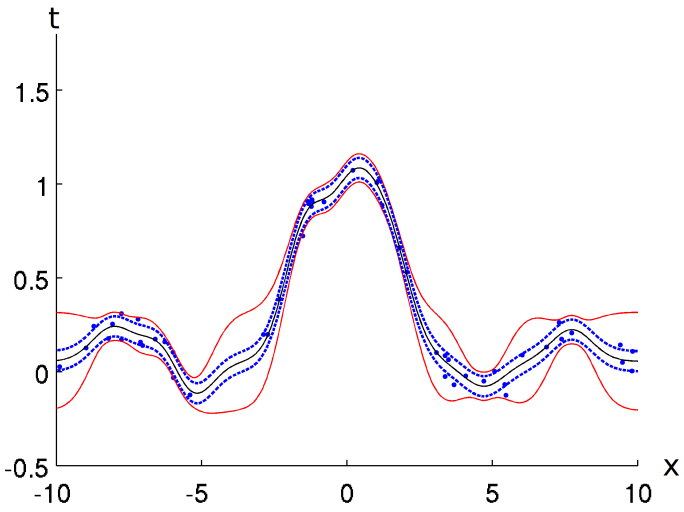

Figure 25: Predictive distribution for sinc function: predictive mean (black solid line), predictive uncertainty (mean ±1 standard deviation, red solid line), isotropic contribution to uncertainty (mean ±1 standard deviation, blue dashed line)

typically remains. Thus, adaptive sampling based on a sparse model has a tendency to be less 'greedy' and more exploratory than Kriging-like interpolators. However, this is a price that is often justified when computational speed is crucial, such as in BCS, where many models must be updated at every system iteration.

3.3.3 Practical Use of Sparse Bayes and RVM for Predictive Uncertainty

While the sparse Bayesian method is appealing for fast prediction of uncertainty, there are practical difficulties when using existing codes. To clarify naming conventions, "relevance vector machine" refers to a specific type of sparse Bayesian model that uses kernels as its basis function, and a "sparse Bayesian" model is one that uses the ARD principle to reduce the number of basis functions from some initial basis. The most common sparse Bayesian model is an RVM, which is popular because it is useful for regression and classification problems in machine learning and also because of its similarity with a famous, non-Bayesian kernel method called a support vector machine (SVM) [150].

There are difficulties in using the pure, kernel-based RVM for adaptive sampling.

The kernel functions typically take the form of radial basis functions that depend on the distance between two points (there is a further description of kernels in the Appendix). For example, the Gaussian kernel has the form

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{D^2}{c^2}\right) \tag{58}$$

where D is the Euclidean distance $\|\mathbf{x}_i - \mathbf{x}_j\|$ and c is a length scale parameter. Conventionally, kernels are centered at data points. An issue arises because kernel values decay rapidly as the distance D increases. Recall that the predictive variance arises because of uncertainty in the model weights $\mathbf{w}$ for linear models $\mathbf{w}^T\boldsymbol{\phi}$, where $\boldsymbol{\phi}$ is the kernel function in this case. No matter what the posterior predictive weight distribution $p(\mathbf{w}|\mathbf{T})$, if the kernel basis functions at some point x^* have values approaching zero, the predictive variance also approaches zero.

If there is a region in a design space that is very sparsely sampled, the predictive variance should be high to reflect the lack of knowledge. However, because points in the region are far from any kernel basis centers, the variances will be very low — this is exactly the opposite of what is intuitive and desirable. This undesirable, low variance is not a failure of the Bayesian approach, but rather the inability to express high uncertainty in that region due to the limited form of the basis functions. To choose a set of basis functions is to select a handful of ways in which observed data can be explained. Common kernel functions of the form of Eq. 58 are limited in their ability to represent high uncertainty in unsampled regions. This results in overconfident predictions in those regions.

This problem is also described in Quinonero *et al* [117]. The authors propose a technique of normalization and decorrelation of the basis functions to allow RVM to show higher uncertainty in unsampled regions. However, the present thesis uses a much simpler method: parametric functions are included in the basis. These functions may include terms like x, x^2, $\sin(x)$, where x is an element of a design vector $\mathbf{X}$. The simplest basis term would be the constant bias, 1. Inclusion of these terms allows

the Bayesian model to express uncertainty through its posterior weights $p(\mathbf{w}|\mathbf{T})$ even in unsampled regions. Even the addition of the bias term is sufficient in many cases to correct the overconfidence in these regions. For example, Figures 26 and 27 show surface plots of RVM predictive standard deviation where blue is low and red is high.

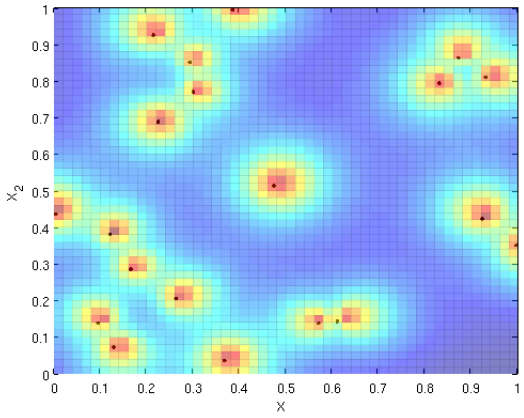

Figure 26: Predictive variance with no bias

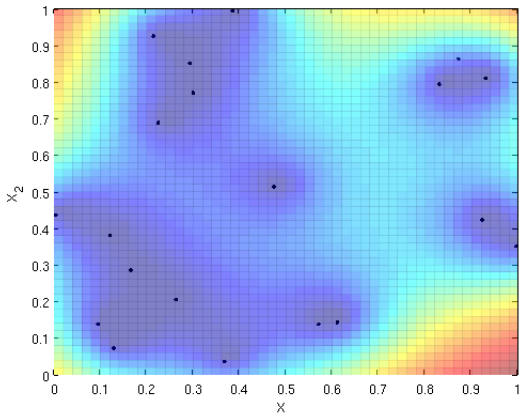

Figure 27: Predictive variance with bias

There is no bias basis function 1 in the first case, but there is in the second case. It can be seen in the second case, uncertainty increases with distance from observed data points, while the opposite is true of the case without a bias.

Strictly speaking, an RVM is a sparse Bayesian model with only kernel functions; conventionally, an RVM with a bias is still called an RVM. In this thesis, this naming convention is followed.

In addition to a bias term and parametric basis functions, the strength of influence of a kernel in unsampled regions can be affected by kernel form. Recalling the Gaussian kernel form, consider the general form

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left[-\left(\frac{D}{c}\right)^p\right] \tag{59}$$

where $D = \|\mathbf{x}_i - \mathbf{x}_j\|_2$, c is a length scale, and exponent p is a tunable parameter. $p = 2$ is the Gaussian case, and $p = 1$ is the Laplace kernel. As p decreases, the kernel function becomes more peaked at its center where $D = 0$. With a sharply peaked kernel centered on a data point, the influence of the data point decays sharply with distance. For this reason, most of the examples in this thesis use a Laplace kernel with $p = 1$ or $p = 0.5$. Combined with the parametric basis functions such as 1, x, x^2, x^3, $sin(\pi x)$, and $sin(2\pi x)$, these sharply peaked kernels tend to show high confidence near observed points and low confidence in unobserved regions, satisfying an intuitive view of uncertainty.

Note once again: the models used in this thesis are generalized sparse Bayesian linear models, which include both kernel functions (like RVM) and parametric functions. Simple RVM with only common kernel functions may not perform well in the BCS application.

3.4 Criteria for Sequential Learning

After discussing sparse Bayesian modeling methods, the focus shifts to how their predictive distributions can be used with for adaptive sampling. Chapter 2 mentioned expected improvement and goal seeking/target matching criteria. These are explained in greater detail here.

3.4.1 Expected Improvement

The expected improvement can be computed in terms of the posterior predictive distribution from a sparse Bayesian model. Recall that the predictive distribution is

given by Eq.(45) through (49):

$$p\left(t^*|\,\mathbf{x}^*,\mathbf{X},\mathbf{T}\right)\sim\mathcal{N}\left(t^*(\mathbf{x}^*)|\,y(\mathbf{x}^*),\sigma^2(\mathbf{x}^*)\right)$$

where

$$y(\mathbf{x}^*)=\mathbf{m}^T\boldsymbol{\phi}(\mathbf{x}^*)$$
$$\sigma^2(\mathbf{x}^*)=\beta_{MP}^{-1}+\boldsymbol{\phi}\left(\mathbf{x}^*\right)^T\mathbf{S}\boldsymbol{\phi}(\mathbf{x}^*)$$

and

$$\mathbf{S}=\left(\beta_{MP}\boldsymbol{\Phi}^T\boldsymbol{\Phi}+\mathbf{A}\right)^{-1}$$
$$\mathbf{m}=\beta_{MP}\,\mathbf{S}\,\boldsymbol{\Phi}^T\mathbf{T}$$
$$\mathbf{A}=\mathrm{diag}(\alpha_{_{MP,i}})$$

If $t_{\min}$ is the minimum (or best) response observed so far, we can define improvement as

$$I(\mathbf{x}^*)\equiv\begin{cases}t_{\min}-t^*(\mathbf{x}^*) & \text{if } t^*<t_{\min}\\ 0 & \text{otherwise}\end{cases}\tag{60}$$

The expected improvement is the centered first moment of $I(\mathbf{x}^*)$ over the predictive distribution of the response $t^*(\mathbf{x}^*)$:

$$\begin{aligned}E\left[I(\mathbf{x}^*)\right]&=\int_{-\infty}^{\infty}I(\mathbf{x}^*)p\left(\xi|\,\mathbf{x}^*,\mathbf{X},\mathbf{T}\right)\mathrm{d}\xi\\&=\int_{-\infty}^{t_{\min}}\left(t_{\min}-\xi\right)\,p\left(\xi|\,\mathbf{x}^*,\mathbf{X},\mathbf{T}\right)\mathrm{d}\xi+\int_{t_{\min}}^{\infty}(0)\,p\left(\xi|\,\mathbf{x}^*,\mathbf{X},\mathbf{T}\right)\mathrm{d}\xi\\&=\int_{-\infty}^{t_{\min}}\left(t_{\min}-\xi\right)\frac{1}{\sqrt{2\pi\sigma^2(\mathbf{x}^*)}}\exp\left\{-\frac{\left(\xi-y(\mathbf{x}^*)\right)^2}{2\sigma^2(\mathbf{x}^*)}\right\}\mathrm{d}\xi\end{aligned}\tag{61}$$

where ξ is a dummy variable standing in for $t^*(\mathbf{x}^*)$. Integrating by parts, the solution is:

$$\begin{aligned}E\left[I(\mathbf{x}^*)\right]=\left(t_{\min}-y(\mathbf{x}^*)\right)&\left[\frac{1}{2}+\frac{1}{2}\mathrm{erf}\left(\frac{t_{\min}-y(\mathbf{x}^*)}{\sigma\sqrt{2}}\right)\right]\\&+\sigma\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{\left(t_{\min}-y(\mathbf{x})^*\right)^2}{2\sigma^2}\right].\end{aligned}\tag{62}$$

3.4.2 Goal Seeking

The goal-seeking (or target-matching) criterion can also be calculated from the posterior predictive distribution of Eq.(45). Let t_{target} the target response. The probability of target match is the predictive distribution integrated over the integral $t_{\text{target}} \pm \epsilon$, where ϵ is some small tolerance like 10^{-5}. This integration has a simple closed form because the predictive distribution is Gaussian.

$$\begin{aligned} P\left(t_{\text{target}} - \epsilon < t^*(\mathbf{x}^*) < t_{\text{target}} + \epsilon\right) &= \int_{t_{\text{target}}-\epsilon}^{t_{\text{target}}+\epsilon} p\left(\xi |\, \mathbf{x}^*, \mathbf{X}, \mathbf{T}\right) \mathrm{d}\xi \\ &= \frac{1}{2}\left[\operatorname{erf}\left(\frac{t_{\text{target}} + \epsilon - y(\mathbf{x}^*)}{\sigma\sqrt{2}}\right) - \operatorname{erf}\left(\frac{t_{\text{target}} - \epsilon - y(\mathbf{x}^*)}{\sigma\sqrt{2}}\right)\right] \end{aligned} \tag{63}$$

3.4.2.1 Caveat: Alternate Forms of Goal Seeking Criterion

In some problems, the target is $t_{\text{target}} = 0$. This is the case in BCS when adaptively sampling the system target discrepancy J_k in BCS and trying to drive it to zero. One can use goal seeking criterion as

$$P\left(0 - \epsilon < J_k(\mathbf{x}^*) < 0 + \epsilon\right) \tag{64}$$

Alternate forms of Eq.64 may be used if different forms are used for the target discrepancy. In early trials of BCS, J_k^2 or even $\sum_k log J_k^2$ were minimized instead of driving J_k to zero. In these early trials (described in Chapter 5), a slightly different form of Eq.64 was used to reflect a BCS formulation of driving J_k^2 rather than J_k to zero.

$$P\left(J_k^2(\mathbf{x}^*) < \epsilon\right) \tag{65}$$

Ultimately, there was no major performance advantage in using the form in Eq.65,

so the simple Eq.64 was adopted for the later test cases in this thesis. The details are reported in each test case.

CHAPTER IV

RESEARCH METHODOLOGY

4.1 Research Questions, Hypotheses, and Predictions

This dissertation investigates a practical solution for a common problem faced by designers. It aims to craft an engineering method rather than to explain a phenomenon or discern a physical law. However, the thesis also looks at the general concept of adaptive sampling for multidisciplinary design — of which BCS is a proof-of-concept example. So, while having a practical goal of improving designers' tradecraft, the thesis follows a scientific procedure. The following subsections describe three sets of research questions, hypotheses and predictions to test.

First, a very general research question, hypothesis, and prediction are stated to frame the entire thesis. These are deliberately general in order to set a research agenda rather than to pose specific tests.

Research Question 1: (General)

How can high fidelity analyses be used for multidisciplinary design if they are too expensive to create a globally accurate surrogate model or to perform direct optimization?

Hypothesis 1: (General)

A Bayesian adaptive sampling method can concentrate sample points in 'favorable' regions of the design space with feasible, low objective values and interdisciplinary compatibility. It can do this by sampling unfavorable regions only enough to establish that they are probably unfavorable.

Prediction 1: (General)

At least one such adaptive sampling method can be demonstrated in a practical multidisciplinary problem.

Thus, the strategy is to show one practical, working example of the general approach. Whether the BCS is the most refined, ideal implementation is not the point. More basically, can the idea of adaptive sampling based on an objective as well as compatibility be done at all? This first set will naturally be answered during the course of studying later, detailed questions.

The second research question/hypothesis/prediction is concerned with the BCS method in particular and defines metrics for its performance.

Research Question 2:

Does the particular method, Bayesian collaborative sampling (BCS), serve as an example of the general sampling strategy described in Hypothesis 1?

Hypothesis 2:

The Bayesian collaborative sampling (BCS) method behaves as in Hypothesis 1 – it concentrates sample points in regions of favorable objective and interdisciplinary compatibility.

Prediction 2:

In a representative design problem, BCS shows higher certainty about a global, compatible optimum than a typical DoE in a test problem. Assume that Bayesian models of the system objective $t(\mathbf{x})$ are fit to DoE and BCS samples of equal size, so that each gives a predictive distribution $P(t(\mathbf{x}))$. A compatible global optimum $\hat{t}(\hat{\mathbf{x}})$ is also independently calculated by optimization with direct fixed point iteration. Let $\delta_{95\%}$ be a measure of Bayesian confidence defined by the probability

$$P\left[\hat{t} - \delta_{95\%} < t\left(\hat{\mathbf{x}}\right) < \hat{t} + \delta_{95\%}\right] = .95 \tag{66}$$

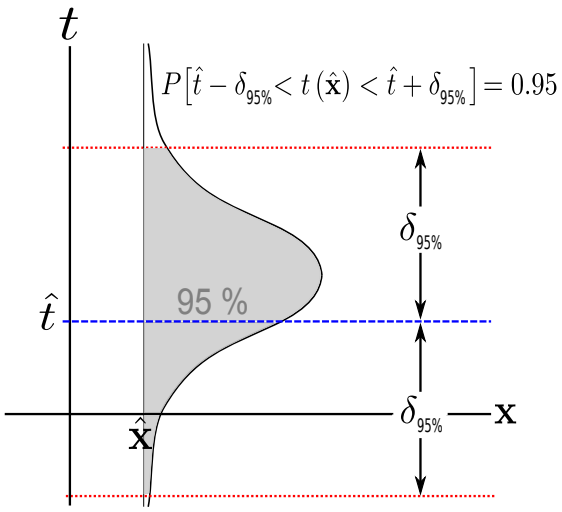

Figure 28: $\delta_{95\%,\,\mathrm{DoE}}$ as a measure of Bayesian confidence at the optimum

as shown in Fig. 28. Then, $\delta_{95\%,\,\mathrm{BCS}} < \delta_{95\%,\,\mathrm{DoE}}$. Lower $\delta_{95\%}$ is better; it indicates more certainty at the optimum.

There are many other possible metrics to test whether BCS operates as intended, but this prediction will adequately capture the essential features of concentrating samples in the favorable, compatible region. The metric $\delta_{95\%}$ will penalize both high confidence in inaccurate mean predictions and low confidence about accurate means.

Note that an expensive direct optimization with FPI is required to provide the reference point to evaluate the metric and compare BCS and a DoE. Therefore, the test of Prediction 2 must be done on a realistic but relatively inexpensive test problem – a subsonic glider wing. Even though the BCS method is intended for efficient use of expensive high fidelity analyses, a much cheaper analysis is needed to allow direct optimization to provide a reference point for studying BCS performance.

At this point, there is a general methodological challenge that affects many MDO research problems. For the researcher, there is a danger of unintentionally tuning optimizers or sampling methods to the test problem. Despite the best intentions of the researcher, it is nearly impossible to make this test a truly unbiased and representative measure of a method's general performance. First, because of the complexity of

realistic engineering problems, most researchers can only afford to focus on one or two design applications. With such a small sample, it is difficult to generalize a method's performance to entire classes of MDO problems. Second, it is very difficult to avoid tuning the test problem to the method. While implementing a method in a practical computer code, many arbitrary parameters must be set and errors must be debugged. Design variable ranges must be selected, and numerical solver parameters must be set. If initial tests fail, the code must be debugged and improved over many trials.

The problem is that the computer implementation unintentionally becomes tuned to the test problem. A common error among MDO researchers is to use, for example, 10,000 function calls to test and debug an algorithm and then only count the 500 iterations of a final test. The true performance of the MDO method actually depended on 10,500 iterations with human intervention.

Therefore, the reader should be wary of interpreting good performance of BCS in the test problem as a promise of good performance in general. There is no real surprise if it performs well; the test problem was deliberately tuned to perform well. This is methodologically reasonable because the main focus of the second hypothesis and prediction is whether BCS works at all and whether it works for the hypothesized reasons. The hypothesis is a falsifiable proposition with a limited, testable prediction.

It will be shown that a test case supports Hypothesis 2. Hypotheses cannot be proved, and single cases usually do not provide strong evidence. However, since it is very unlikely that the prediction could be confirmed if the hypothesis were untrue, then even the single case raises one's confidence in the hypothesis. For example, suppose that one hypothesizes a new thermodynamic cycle. If one builds a working engine that demonstrably operates on that principle, confidence in the thermodynamic cycle increases substantially since it is unlikely that a complicated engine works by mere chance. Similarly, even a single case of working BCS can raise confidence in its

underlying principle.

If one is reasonably confident that BCS works as hypothesized, the next task is to characterize its function. At first glance, an obvious test might be to compare its performance with many other benchmark methods. However, such a test suffers from the same difficulties as in Hypothesis 2. It is almost impossible to avoid tuning the algorithm to the test problem. Most MDO or sampling methods contain arbitrary parameters that must be set. These include relaxation factors, choices of internal solvers, variable ranges, convergence tolerances, warm start sample size. Often, the performance of these algorithms depend heavily on these arbitrary choices. If BCS is compared to another method, the results of such a contest may be reversed by small changes in these settings.

This was indeed the case in early experiments. A Bayesian expected improvement (EI) adaptive sampling algorithm with simple looping fixed point iteration (FPI) was used on the glider test problem as a benchmark. The FPI proved to be extremely expensive, often requiring several hundred discipline function calls to achieve compatibility for a single design. In some cases, the EI/FPI method would become trapped in FPI in which coupling variables would oscillate indefinitely. By any metric, BCS was more efficient. To provide a more reasonable benchmark, a relaxation factor was introduced to the FPI and variable ranges were adjusted. This too was unreliable and expensive, so an adaptive relaxation factor was also introduced, and convergence tolerances were further adjusted. Finally, after tuning with over $O(10^5)$ function calls, the benchmark method functioned reliably. But because of this high dependence on human-in-the-loop tuning, it is difficult to make pronouncements that BCS or EI/FPI is "better."

This difficulty in comparing optimization or sampling methods is a serious and general methodological problem in MDO research. The best option would be to create a large test suite of different engineering examples. This would alleviate the

problem of over-tuning a method to an example and the resulting credibility problem. If a method were tuned to perform well on a dozen test problems without individual adjustment, the tuning would take on the more benign character of a heuristic rule of thumb. There would be some confidence that the same rule of thumb would lead to good performance on a new application. Unfortunately, the MDO research community currently lacks an authoritative, common test suite, and it is well beyond the scope of a single thesis to develop one.

Despite this difficulty, it is still possible to ask well-posed questions to characterize the performance of BCS. The designer's general question is "when should I use this method?" First, qualitative reasons for using BCS have been described. Recall that the decoupling of disciplines means that BCS is useful when communication between discipline designers is difficult or even undesirable. For example, an engine company and an airframe company may wish to cooperate more easily but without sharing propietary analysis codes. The bi-level decomposition makes it easier to preserve disciplinary autonomy; existing expertise and best practices in each discipline can be preserved without interference or excessively tight control from a central leadership. A designer may become interested in BCS for these reasons, but wonder if there are cases where BCS has disadvantages that outweigh these benefits. This thesis will investigate one important mechanism that may make BCS perform badly. Specifically, BCS may become less useful as *coupling bandwidth* between disciplines increases.

Coupling bandwidth refers to the number of coupling variables passed between disciplines. This is distinct from coupling strength, which is the influence of input coupling variables on the output of a discipline. The bi-level problem decomposition in BCS or the original collaborative optimization requires that coupling variables become system level target variables. The purpose of this bi-level decomposition is to avoid costly fixed point iteration. But the main cost is a larger system design space, which is subject to the "curse of dimensionality."

It is not possible to generally characterize this tradeoff, and the answer is probably problem-dependent. For now, a more modest goal is to study whether this phenomenon occurs and if it is significant in a representative design problem. A simple EI/FPI benchmark is also run alongside BCS as coupling variables are increased in the same glider test case used in Hypothesis 2. The purpose of the benchmark is not to study say whether one method works better. For stability and reliability, the benchmark method is heavily tuned to the glider problem, so such a comparison may be misleading. The real purpose is to show how relative performance changes as coupling bandwidth changes. For this, the benchmark will provide some reference for the effect of unavoidably adding physical complexity to the test case. Coupling variables are added to the glider example by increasing the resolution of load and deflection information passed between disciplines. This additional resolution and physical complexity might degrade BCS performance independent of the coupling bandwidth effect. The EI/FPI benchmark may help to screen out the effect since it is also affected by the greater physical complexity. To foreshadow results in later chapters, this EI/FPI benchmark proved to be too noisy and patternless to be very useful, but it only serves an optional or subsidiary role in the following research questions and hypotheses.

The third research problem is described below. Note once again that it characterizes one important mechanism by which BCS may perform worse, rather than trying to generally compare BCS performance to other methods. It focuses on coupling bandwidth, which is the number of coupling variables.

Research Question 3:

What are major factors that affect BCS performance?

Hypothesis 3:

Coupling bandwidth has a major effect on BCS performance. As the number of of

coupling variables increases, the system level of BCS increases in dimension and incurs the "curse of dimensionality."

Prediction 3:

BCS performance degrades as coupling bandwidth increases in one engineering test problem. Performance is measured by $\delta_{95\%,\,\mathrm{BCS}}$ and $\epsilon_{95\%,\,\mathrm{BCS}}$. $\epsilon_{95\%,\,\mathrm{BCS}}$ is measured by

$$P_{\text{compatibility}}(\epsilon_{95\%}) \approx \prod_{k}^{K} P\left(-\epsilon_{95\%} < J_k(\mathbf{Z}) < \epsilon_{95\%}\right) = 0.95 \tag{67}$$

for system design $\mathbf{Z}$ and target discrepancies J_k for $k = 1, 2, \ldots, K$ coupling variables. The experiments are also run on EI/FPI for reference.

The product in Eq.67 is the probability that the system design $\mathbf{Z}$ will result in all K target discrepancies J_k simultaneously falling within an $\epsilon_{95\%}$ tolerance of zero, assuming that all J_k are independent. In practice, this value can be computed as follows. After Bayesian models are regressed to $J_k(\mathbf{Z})$, a Gaussian predictive mean m_k and variance s_k^2 can be computed. Given some tolerance ϵ, the probability of compatibility can be computed by:

$$P_{\text{compatibility}}(\epsilon) = \prod_{k}^{K} \left\{ \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{\epsilon - m_{k,\text{predictive}}}{\sqrt{2s^2_{k,\text{predictive}}}}\right)\right] - \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{-\epsilon - m_{k,\text{predictive}}}{\sqrt{2s^2_{k,\text{predictive}}}}\right]\right)\right\} \tag{68}$$

Eq. 68 is varied with respect to ϵ to solve for $P_{\text{compatibility}}(\epsilon_{95\%}) = 0.95$.

Finally, recall that the original motivation behind BCS is the efficient use of expensive, high fidelity analysis for early design in highly coupled problems. Perhaps the obvious, final test is to demonstrate BCS in such an expensive, coupled problem. For this, a formal hypothesis and prediction will not be stated, for the question is inherently open-ended.

Research Question 4:

Can BCS be implemented practically with high fidelity codes like CFD?

If BCS works with inexpensive codes, it may seem obvious that it will likely work with expensive codes. However, there are many new challenges for high fidelity. For example, if the disciplinary analysis requires numerical solution of discretized differential equations as with CFD, the size of residuals compared to BCS compatibility tolerances may affect the function of BCS. Or, for example, there may be more or less efficient ways to reuse volume meshes or converged solutions from previous simulations to accelerate new simulations. The last research question is therefore left open-ended to allow for serendipitous discoveries and to learn practical lessons rather than to pose formal tests. Because of the expense of high fidelity codes, it is not possible to do thorough comparisons with other methods or with an optimum. For example, the previous hypotheses involved certainty at the optimum, so the problem must actually be optimized at considerable expense. In contrast, a high fidelity demonstration need not involve such expensive tests and is only concerned with practical implementation.

For this purpose, the Boeing/NASA N2B boundary layer ingesting hybrid wing body was originally selected as the test case. The example was well suited for a BCS demonstration because the concept inherently relies on coupled aero-propulsion physics, which in turn may require high fidelity CFD even for early sizing decisions. However, during the course of thesis research, CFD sensitivity studies showed that analysis cost for the boundary layer ingesting aircraft was unexpectedly high. It became doubtful that BCS could be successfully demonstrated within the thesis's computing budget. Instead, a much simpler example was used: aero-propulsion design of an axisymmetric turbojet engine nacelle. Despite this, the incomplete N2B case is still important because it reveals lessons and pitfalls during high fidelity BCS implementation. This experience is important to document for future researchers who may study similar cases for multidisciplinary sampling. Both the incomplete

N2B and the engine nacelle will be discussed in detail in Chapter 6.

BCS is implemented in MATLAB for these case studies. For the Bayesian model, a sparse Bayesian code was written in Python for early studies. However, the NASA Ames supercomputing cluster used for this thesis did not support the SciPy scientific computing modules required by the Python code. So for large scale tests, the BCS code instead relies on a modified MATLAB-based code written by the relevance vector machine's (RVM) original author. This code is available under GNU public license from Michael Tipping; a description of this code is given in [19], [149], and [150]. An additional benefit of using the author's public code is that it makes BCS more transparent and credible, especially since much of previous literature on RVM performance is based on this code. The code is modified to include parametric basis functions in addition to the kernel functions and to output posterior predictive variances. The model is therefore a general sparse Bayesian linear model rather than a kernel-based RVM.

BCS was first tested on simple test functions, then low-fidelity wing design, and finally for high fidelity engine nacelle design.

4.2 Algebraic Test Problem

For Hypothesis 2 (concerned with the basic function of BCS), the first set of test problems used algebraic expressions for system and discipline-level functions. Typically, MDO studies often focus on canonical algebraic test functions for initial development. Unfortunately, these experiments are of limited use for BCS because it was found that they do not test the Hypothesis 2 predictions in a realistic manner. The very smooth algebraic functions often allow Bayesian kernel models to fit with very small uncertainty, often with isotropic noise variances close to zero ($< 10^{-6}$). In such cases, the adaptive learning algorithm behaves as a optimizer, and BCS reduces to an efficient variant of the original CO method. Of course, the original test was to

compare BCS and a DoE in their performance as sampling schemes. This comparison sometimes yields spectacular results for BCS, but the manner of these successes does not generalize well to more realistic, noisy engineering analysis codes.

One of these algebraic examples is discussed here. Note that this particular example and its good results are not supposed to be representative, but rather to show basic BCS behavior and also to demonstrate why such algebraic problems are not suitable tests.

The system-level objective function F_{sys} is a modified version of the well-known Rastrigin test function (the "egg-carton" function) in three dimensions. The two discipline-level functions a_1 and a_2 are functions of local design variables x_1 and x_2, respectively, as well as a shared/global design variable s_1 and the output of the other discipline.

$$
\begin{aligned}
&\text{Minimize:} && F_{sys} && = a_1^2 + a_2^2 + s_1^2 + cos(4a_1) + cos(4a_2) + cos(4s_1) \\
&\text{with respect to:} && && x_1, \quad x_2, \quad s_1 \\
&\text{where:} && a_1 && = a_1(a_2, s_1, x_1) \\
& && a_2 && = a_2(a_1, s_1, x_2) \\
&\text{subject to:} && a_1 && = 2x_1^2 - x_1 - a_2 - s_1 - 2 \\
& && a_2 && = 2x_2^2 - x_2 - a_1^2 + s_1^2 - 2 \\
& && g_1 && = x_1 + a_2 - 2s_1 + 3 \leq 0 \\
& && g_2 && = 2x_2 - a_1 - s_1 + 3 \leq 0
\end{aligned}
$$

Figure 29 shows a design structure matrix for the test problem. Note the feedback from a_2 to a_1. This loop requires fixed point iteration to ensure that $a_{2,\text{guess}}$ that is an input to the a_1 function converges to $a_{2,\text{actual}}$ output from the a_2 function.

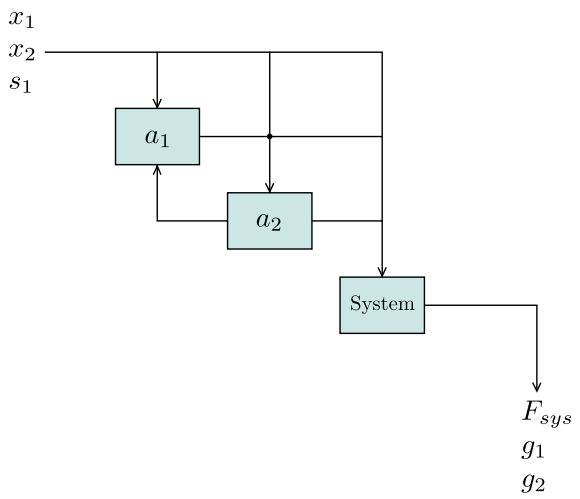

Figure 29: DSM of algebraic test problem

4.2.1 Results

The original intent of the experiment was to produce samples with 500, 1000, 2000, and 5000 function calls using a Latin hypercube DoE and BCS. The same type of Bayesian model would be fit to both DoE and BCS samples. The certainty of each model would be compared at a known optimum. The optimum response of 0.95 was found by running over 60 trials of genetic algorithms (GAs) with different population sizes and MDO architectures. However, the BCS method actually converged on the optimum in 8 of 10 trials and was several orders of magnitude faster than the GAs. This means that there is little benefit in comparing its predictive accuracy at that point with a DoE-based model, since the point is known with very high certainty.

Both the smoothness and low order of the functions mean that they are closely approximated by the Gaussian kernels used in the sparse Bayesian models. Because the noise precision β estimated by the model is relatively high (i.e. the variance is low), most of the design space can be ruled out quickly and with high certainty, causing BCS to rapidly converge on an optimum without having to double-check unfavorable regions. The method may not converge on an optimum so readily with analysis codes like CFD or FEA with imperfect discretization and convergence tolerances.

Nonetheless, BCS performance as a global optimizer is worth noting. Four architectures are compared: an individual discipline feasible (IDF) method; a direct fixed point iteration (FPI) method, also known as multidisciplinary feasible (MDF); collaborative optimization (CO); and BCS. The IDF and FPI methods used genetic algorithms in Matlab within their respective architectures. Because the FPI method enforces compatibility directly for every design – and also because the method consistently converged on the best optimum value in all ten trials – the FPI optimizer was assumed to be the "truth model" or reference method.

Table 2 shows comparison with other optimizers where each row represents ten trials for each setting. In the second column, the genetic algorithm population is given for methods that use the optimizer internally. If a method performs poorly, the GA population was varied to check if the results are sensitive to the optimizer. The third column is 'global optimization success fraction,' which is the fraction of ten repeated trials in which the method successfully converged on the global optimum. The last column is the performance metric of most concern to the designer: the number of function calls. It can be seen that BCS converged relatively consistently (80% of the time) with substantially fewer function calls than the other methods. The IDF method tended to converge at local optima and thus has a lower success fraction. The direct FPI method consistently converged on the global optimum, but was relatively

Table 2: Results on algebraic test problem — ten trials each

Method	GA population	global opt success fraction	avg discipline function calls
IDF (GA)	50	20%	2.06×10^4
	100	30%	2.86×10^4
	200	10%	4.72×10^4
	500	20%	1.05×10^5
	1000	40%	2.19×10^5
FPI (GA)	150	100%	1.13×10^5
CO (gradient-based)	—	0%	—
BCS	—	80%	795

costly in terms of function calls. The CO formulation and converged only on local optima. It was suspected that this may be due to its internal use of gradient-based optimizers. However, using genetic algorithms did not improve its performance, but this could be due to the particular implementation in Matlab. The performance may depend on default genetic algorithm parameters such as recombination probabilities, but such issues were not investigated for this minor test.

Figures 30 and 31 show the sampling criterion

$$-\log E\left[I\left(F(\mathbf{Z})\right)\right] - \sum_{h} \log P\left(G_h\left(\mathbf{Z}\right) \leq \epsilon\right) - \sum_{k} \log P\left(J_k^2\left(\mathbf{Z}\right) \leq \epsilon\right) \tag{69}$$

plotted against system variables $\mathbf{Z} = [a_1\ a_2]$. The dark dimple (circled) in the low

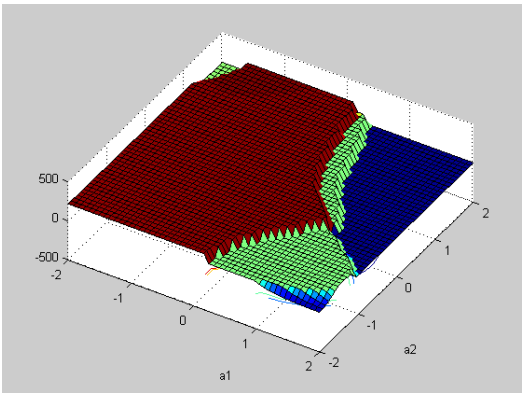

Figure 30: Sampling criterion after 20 system iterations

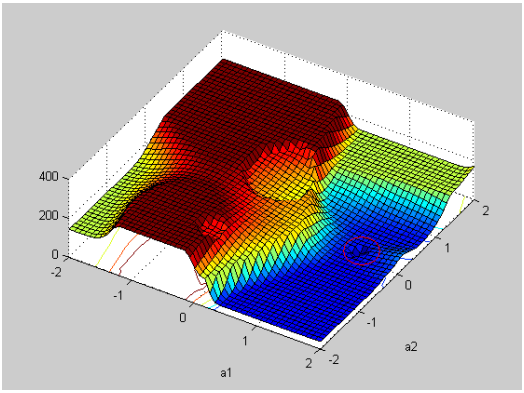

Figure 31: Sampling criterion after 70 system iterations

region of Figure 31 contains the global optimum. The raised plateaus are regions that are infeasible, have a high objective value or have low probability of compatibility.

Other informal tests were also done with other test functions at the system level: the Branin function, Shekel function, and the Rosenbrock "banana" function. Polynomials of varying order were used for the disciplinary functions. The number of disciplines, system and disciplinary variables, coupling variables, and constraints were varied. In all of these cases, sampling proceeds until the estimated isotropic noise variance becomes close to zero, and BCS behaves as an EGO-type optimizer.

This low estimated isotropic variance (or high precision β) suggests the underlying reason for this seemingly excellent performance of BCS. The smoothness of the analytic test functions and the similarity of the basis functions to them allowed BCS to quickly fit accurate approximations and optimize them. For example, suppose that the BCS Bayesian models use polynomial bases and the actual test function is a paraboloid. Because the bases would be fit with very little error, and BCS would converge near the true optimum in a very small number of iterations. For this reason, smooth test functions give a very unrealistic view of how BCS would work in a realistic engineering problem. Typical engineering codes are "noisy," with internal solver tolerances and roundoff errors. For a more realistic test of BCS, a low speed glider wing test problem is investigated in the next chapter.

CHAPTER V

GLIDER WING TEST PROBLEM

Hypothesis 2 is concerned with whether BCS works at all – will it concentrate sample points in regions of favorable objectives and probable compatibility? It was found that smooth mathematical test functions give unrealistically favorable results for BCS because the form of the Bayesian model basis functions are able to closely match the shapes of the test functions. Therefore, the next test problem is a realistic engineering test case using low fidelity analysis tools. While simple, they have internal solver tolerances and discretized differential equations and can therefore represent the non-smoothness typical of many engineering analysis codes. Most of the theory and performance tests in this thesis are performed on this low fidelity example.

The problem is the aerostructural design of a small glider wing using potential flow with boundary layer modeling and the Euler beam equation. The glider wing was chosen because its physics are in regimes where the low fidelity, linear models are most accurate. For example, the wings tend to be thin, high aspect ratio wings in low subsonic flow where vortex lattice methods tend to most accurate. Likewise, a long, thin spar with small deflections is accurately modeled by Euler beam theory. A simple yet reasonable construction method was assumed so that wing weight could be estimated and checked with physical examples. Despite the simplicity of the example, it was expected to have enough noisy features due to discretization and non-linear physics due to boundary layer effects so that it would provide a realistic test of BCS versus a DoE.

This chapter will first give a detailed description of the unmodified multidisciplinary test problem. Next, it will give a step-by-step description of the implementation of BCS following four major steps: 1) bi-level formulation, 2) setup of analysis codes, 3) initialization, and 4) adaptive sampling. Then, results of performance tests are given to address the research hypotheses.

5.1 Glider Wing Design Problem Statement

The design problem is stated as follows.

Minimize:	F_{sys}	$= -L/D_{\max}$, maximum lift to drag ratio
with respect to:		aspect ratio
		camber
		thickness/chord ratio
		chordwise location of maximum camber
		chordwise location of maximum thickness
		leading edge radius
		root sparcap width
		cruise speed
subject to:	g_1	$=$ sinkrate $- 1.0$ ft/s ≤ 0
	g_2	$=$ deflection$_{\max} - 0.1$ span ≤ 0
	g_3	$=$ sparcap stress$_{\max} - 0.5\,\sigma_{\text{yield, carbon}} \leq 0$
given:		1 lb aircraft weight without wing
		10 ft span
		150 lbf maneuver load at 120 ft/s
		steady level flight for $L/D_{\max}$

Fig. 37 shows the design structure matrix for the design problem.

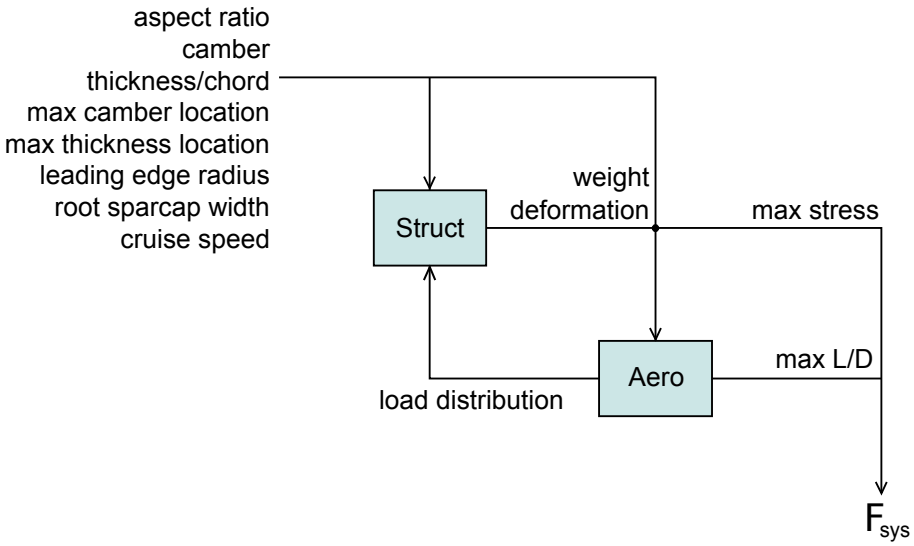

Figure 32: Wing design structure matrix

The coupling variables are weight, load distribution, and deformation. The load distribution and deformation are computed during a 150 lbf lift maneuver. Only loads and deformations in the transverse direction (orthogonal to span and free stream) are considered. Load and deformation distributions are also assumed to vary only spanwise and act along a single wing spar. Inertial effects due to the mass distribution along the wing are ignored. This simple load scenario actually corresponds to a radio controlled glider being launched with a towline and winch. To further reduce coupling bandwidth, a polynomial equation is fit to the load distribution so that the load can be expressed in terms of three weighting coefficients. Likewise, the deformation is expressed as two transverse deflections at each of two spanwise stations.

Three inequality constraints are given, and these are enforced as penalties on the objective function. The sinkrate constraint is included because early tests showed that unconstrained minimization of the L/D ratio leads to very thick wings and wide spar caps. The physical reason is that the high weight acts as ballast so that the glider achieves high L/D ratios at high speeds and Reynolds numbers. This is unrealistic because most gliders or sailplanes do not merely cruise at optimal glide angles but also have minimum sink tasks. Therefore, a 1.0 ft/s sink rate constraint is imposed.

The wing tip deflection constraint is imposed because the linear beam theory model is unreliable for high deflections. The maximum sparcap stress constraint is imposed with the simplifying assumption that the spar cap will not buckle but will fail in spanwise compression. A safety factor of 2 is therefore imposed with constraint g_3.

Because of the feedback in Figure 32 from the aerodynamics discipline to the structures discipline, the design problem implicitly requires a set of compatibility conditions:

$$h_1 = \text{structural weight} - \text{lift} = 0$$

$$h_{(1+i)} = \text{deformation parameter}_{i,\text{aero}} - \text{deformation parameter}_{i,\text{structures}} = 0$$

$$\text{for } i = 1, 2$$

$$h_{(4+j)} = \text{load distribution parameter}_{j,\text{aero}} - \text{load distribution parameter}_{j,\text{structures}} = 0$$

$$\text{for } j = 1, 2, 3$$

In the unmodified design problem, these equality constraints for compatibility must be enforced by fixed point iteration. For each design, all h constraints must be minimized with respect to the coupling variables for weight, deformation and load distribution. If any h cannot be driven to zero (or a small tolerance about zero), then the design is incompatible. This FPI process can be accomplished by simple looping – guessing initial values for the coupling variables and updating them sequentially – or by numerical root finding algorithms. For simplicity, a simple looping method with a manually tuned relaxation factor is used in this thesis as a reference method. In most cases, the FPI costs between 5 and 10 function calls in each discipline, although certain designs may require hundreds of function calls. Again, note that the foregoing describes the unmodified design problem. The problem is later restated as a bi-level problem for BCS to avoid FPI and make efficient use of a limited budget of function calls.

After stating the unmodified glider test problem and selecting the analysis codes, the designer can see in this case that many analysis function calls will be expended in fixed point iteration on the aerodynamic load distribution, weight, and structural deformation. To avoid this cost while making the most of a limited analysis budget, the designer turns to BCS. The following sections step through four major steps in implementing BCS for the glider problem.

5.2 Bi-level Decomposition for BCS

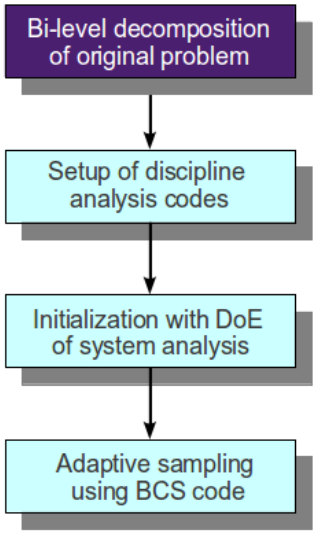

Figure 33: Bi-level decomposition is the first step in implementing BCS

The first step in implementing BCS is to decompose the original problem into a system level and discipline level. There are three sets of variables, for system, aerodynamics, and structures:

$$\mathbf{Z}_{\text{sys}} = \begin{bmatrix} \text{wing area} \\ \text{aspect ratio} \\ \text{camber} \\ \text{thickness/chord ratio} \\ \text{chordwise location of maximum camber} \\ \text{chordwise location of maximum thickness} \\ \text{root sparcap width} \\ \text{cruise speed} \\ \text{weight} \\ \text{load distribution parameter}_1 \\ \text{load distribution parameter}_2 \\ \text{load distribution parameter}_3 \\ \text{deformation parameter}_1 \\ \text{deformation parameter}_2 \end{bmatrix}$$

$$\mathbf{X}^{\text{aero}} = \begin{bmatrix} \text{deformation parameter }_{\text{1,local}} \\ \text{deformation parameter }_{\text{2,local}} \end{bmatrix}$$

$$\mathbf{X}^{\text{struct}} = \begin{bmatrix} \text{weight} \\ \text{load distribution parameter }_{\text{1,local}} \\ \text{load distribution parameter }_{\text{2,local}} \\ \text{load distribution parameter }_{\text{3,local}} \\ \text{root sparcap width} \end{bmatrix}$$

Each discipline has a locally relevant set of system target variables $\mathbf{Z}_{\text{sys}}^{(discipline)}$ and their locally computed equivalents, $\mathbf{Z}_{\text{local}}^{(discipline)}$. For example, there is a $\mathbf{Z}_{\text{local}}^{\text{struct}}$ and $\mathbf{Z}_{\text{sys}}^{\text{struct}}$, where

$$\mathbf{Z}_{\text{local}}^{\text{struct}} = \begin{bmatrix} \text{weight} \\ \text{deformation parameter}_{\text{1,local}} \\ \text{deformation parameter}_{\text{2,local}} \end{bmatrix}$$

and $\mathbf{Z}_{\text{sys}}^{\text{struct}}$ takes the values for the same elements assigned by $\mathbf{Z}_{\text{sys}}$. Note that in this case, the structures discipline has an extra degree of freedom, root sparcap width, which it can optimize to minimize the discrepancy between the system targets and their local equivalents, $\mathbf{Z}_{\text{sys}}^{\text{struct}}$ and $\mathbf{Z}_{\text{sys}}^{\text{struct}}$. On the other hand, the aerodynamics discipline lacks this extra degree of freedom, so the disciplinary problem is simply one of evaluating the target discrepancies.

In summary, the original glider test problem can be restated as a bi-level BCS problem:

SYSTEM LEVEL

$$\textbf{Minimize:} \qquad F = \; -L/D_{\max} + \text{constraint penalties}$$

$$\textbf{with respect to:} \qquad \mathbf{Z}_{\text{sys}}$$

$$\begin{aligned} \textbf{subject to:} \quad Q_{\text{aero}}(\mathbf{Z}_{\text{sys}}) &= 0 \\ Q_{\text{struct}}(\mathbf{Z}_{\text{sys}}) &= 0 \\ g_1 = &\;\; \text{sinkrate} - 1.0 \text{ ft/s} \leq 0 \\ g_2 = &\;\; \text{wingtip deflection}_{\max} - 0.1 \text{ span} \leq 0 \\ g_3 = &\;\; \text{sparcap stress}_{\max} - 0.5\, \sigma_{\text{max, carbon}} \leq 0 \end{aligned}$$

given: aircraft weight without wing

150 lbf maneuver load at 120 ft/s

steady, level flight for $\text{L/D}_{\max}$

by sampling:

$$-\log E\left[I\left(F(\mathbf{Z})\right)\right] - \sum_k \log P\left(J_k^2\left(\mathbf{Z}\right) \leq \epsilon\right)$$

for $k = 1, 2, \ldots 6$ coupling variables

and updating: Bayesian models $p(F|\mathbf{Z}), p(J_k|\mathbf{Z})$

for $k = 1, 2, \ldots 6$ coupling variables

AERODYNAMICS DISCIPLINE

$$\textbf{Evaluate:} \quad Q_{\text{aero}}\left(\mathbf{X}^{\text{aero}}\right) \;=\; \sum_k^{K_{\text{local}}^{(i)}} J_k^2$$

$$\text{for} \quad J_k = Z_{\text{k,sys}} - Z_{\text{k, local output}}$$

STRUCTURES DISCIPLINE

$$\textbf{Minimize:} \quad Q_{\text{struct}}\left(\mathbf{X}^{\text{struct}}\right) \;=\; \sum_k^{K_{\text{local}}^{(i)}} J_k^2$$

$$\textbf{with respect to:} \qquad \mathbf{X}_{\text{local}}^{\text{struct}}$$

5.3 Setup of Discipline Analysis Codes

The next step of implementing BCS is to select discipline analysis codes and arrange them to accept inputs and yield outputs according to the new bi-level problem. The

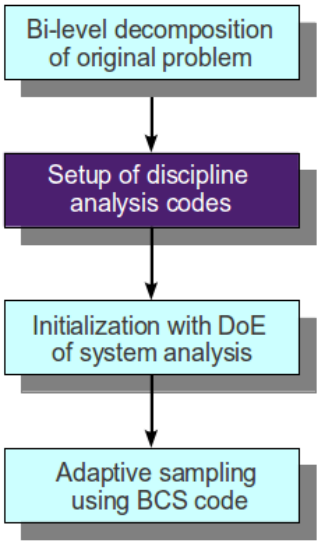

Figure 34: Bi-level decomposition is the first step in implementing BCS

analysis codes for the aerodynamics and structures disciplines are described in detail.

5.3.1 Low Fidelity Aerodynamic Analysis

Aerodynamic analysis is done with Athena Vortex Lattice (AVL) and XFOIL by Mark Drela and Harold Youngren of MIT [42], [40]. Figure 35 shows a sample wing model in AVL. To account for separation and other boundary layers effects, profile drag is computed with XFOIL, which is a two dimensional potential flow solver with boundary layer interaction. After running a sweep of C_l vs. C_d for around 10 points (depending on the number of converged analyses), a parabola is fit to the drag polar by least squares. This polynomial fit is then passed to AVL. This approximation is used to add profile drag to induced drag computed by the vortex lattice code. In other words, AVL is first used to calculate local induced angles of attack and lift coefficients at spanwise stations, and these are in turn used to approximate the local profile drag. This method assumes that profile drag and induced drag are independent, which is

reasonable for high aspect ratio wings. An AG-24 sailplane airfoil is used as a baseline, and thickness, camber, leading edge radius, and other design variables are adjusted in a geometry design module of XFOIL to create new airfoils.

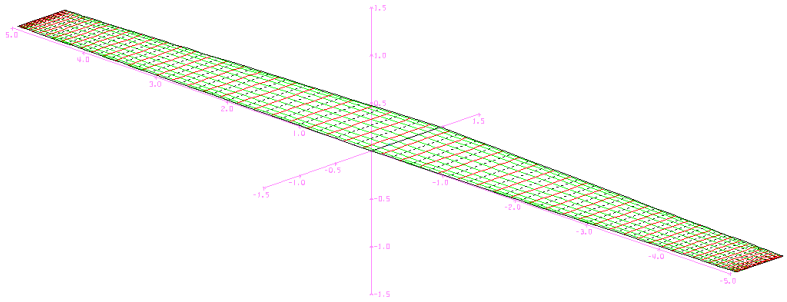

Figure 35: AVL model of high aspect ratio glider wing

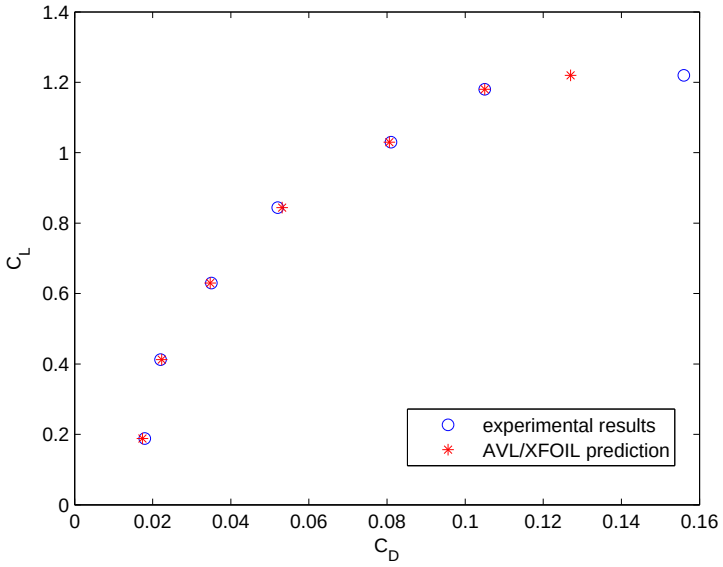

Figure 36: Validation of drag polar — Clark Y airfoil (t/c=12%), AR=6, Re=149000

Figure 36 shows validation data for AVL/XFOIL with NACA wind tunnel tests on a finite wing of similar geometry and flight regime [85]. AVL/XFOIL is expected to be accurate for thin airfoils and high aspect ratio wings, and the BCS test problem uses thickness/chord of 8 to 12% and aspect ratios between 8 to 16. An experimental case was deliberately chosen to be relatively ill-suited to the analysis codes — it has a thick Clark-Y airfoil (12%) and a low aspect ratio of 6. Despite this, AVL/XFOIL matches

wind tunnel data closely. The NACA report does not include enough information to determine uncertainty ranges for the empirical data, but this level of validation may be sufficient to proceed with the BCS test problem. After this validation, however, it was found that the XFOIL code tends to be unstable at low Reynolds numbers and for thick airfoils. In some later tests in this thesis, a neural net surrogate model had to be fit to a large sample of on airfoil variables from XFOIL for stability reasons. The neural nets were fit in Matlab with $O(1000)$ points and validated with 10% random holdout data with model representation error under 5% of the response range. Recall, however, that the AVL/XFOIL combination used in the validation does not use XFOIL directly either and passes a response surface from XFOIL to AVL, so there is no major theoretical difference.

5.3.2 Low Fidelity Structural and Weight Analysis

Structural analysis is done with the Euler-Bernoulli thin beam equation:

$$\frac{\partial^2}{\partial x^2}\left(EI\frac{\partial u^2}{\partial x^2}\right) = w$$

where u is the transverse deflection, E is the elastic modulus, I is the cross sectional area moment of inertia, and w is the applied force per length. The equation is numerically integrated in Matlab. The Matlab calculations were verified by comparison to analytic solutions of well-known examples, such as a uniformly loaded cantilevered beam. Based on these comparisons, the wing spar is discretized into 200 segments.

It is assumed that most of the weight of the aircraft is in the fuselage and that a single spar is the main load-bearing member in the wing. The wing spar is then modeled as a cantilever beam fixed at its root. For further simplification, a sufficient thickness of fiberglass wrap shear web and vertical grain spar core are assumed to ensure that they will not fail before the spar caps. It is assumed that the web will not fail in shear and the spar core will not crush due to transverse compressive loads. Therefore, only the spar caps play a significant role in the structural design of the

wing. Figure 37 shows details of the wing construction.

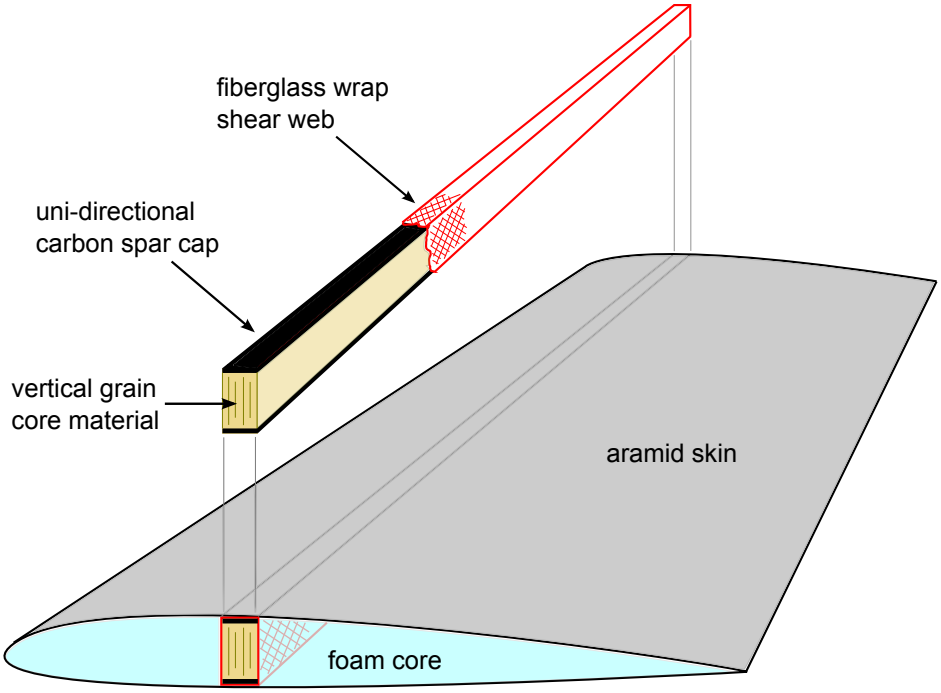

Figure 37: Wing construction

This construction method is also used for structural weight estimations. These estimates assume material properties using common commercially available products. For example, the aramid wing skin is assumed to be a 1.8 oz/yd^2 Kevlar 120 layup in an equal weight of epoxy. The foam core is Dow HiLoad 60 (R) high density polystyrene at 2.2 lb/ft^3, and the spar core is 8.0 lb/ft^3 material like Rohacell(R) or end-grain balsa wood. These assumptions match the construction of several small glider examples included as tutorial cases included with AVL software.

Figure 38 shows the converged geometry and load vectors after fixed point iteration between aerodynamic loads and structural deformation.

The third and fourth steps of implementing BCS are the warm-start and the adaptive sampling, as in Figure 39. However, the details such as warm-start sample size and solver choices are best explained in the context of the formal tests in the following sections. This is because many of the BCS parameters are set in order to give a fair tests. The first test is a comparison of BCS and an off-line DoE. In

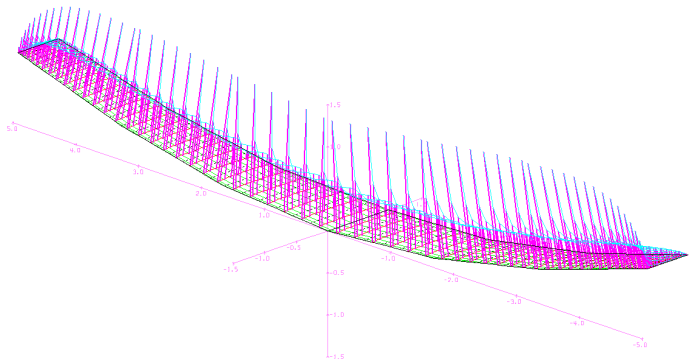

Figure 38: AVL model with converged loading and deflection at maneuver condition

later sections, the BCS implementation is modified further to test individual aspects affecting performance, such as coupling bandwidth or warm-start size.

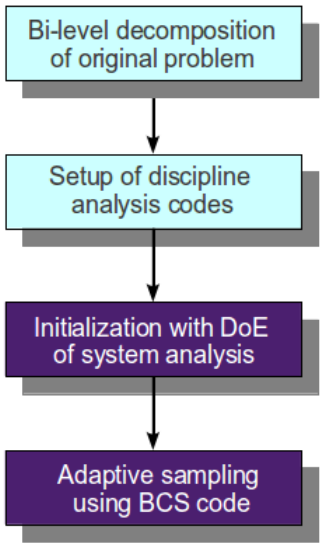

Figure 39: Bi-level decomposition is the first step in implementing BCS

5.4 Comparison of BCS and an Off-Line DoE

Recall that the second hypothesis concerns whether BCS works according to the general strategy of adaptive sampling by a system objective and interdisciplinary compatibility. To this end, the second prediction requires a test: does BCS provide higher certainty at the true, compatible optimum compared to a simple off-line DoE. This is not intended to be a very difficult test of performance; it is rather to show

that BCS works and that it works because of the hypothesized principles. The steps of this test are:

1. Run a simple optimizer on the original problem with FPI to find the true optimum for reference
2. Run a Latin hypercube off-line DoE on the original problem formulation
3. Run BCS on the bi-level formulation
4. Evaluate the predictive uncertainties of BCS and the DoE at the reference optimum

Recall that the predictive uncertainty is measured by $\delta_{95\%}$ defined in the last chapter – a lower value is better, since it indicates a more accurate prediction at the true optimum. If $\delta_{95\%}$ is lower for BCS, this test supports Hypothesis 2: BCS does in fact concentrate points according to a system objective and compatibility.

5.4.1 Implementation Details of BCS vs. DoE Test

The BCS method is compared with a Latin hypercube DoE. The DoE uses fixed point iteration to enforce aerostructural compatibility for each design. Each design point in the DoE requires a number of FPI iterations that is not precisely known beforehand. The DoE is adjusted such that resulting total number of aerodynamics discipline function calls are roughly the same as that used in BCS. Since many aerodynamics function calls are required for every structural function call, a limit of 5000 aerodynamic function calls are used for both DoE and BCS.

The DoE is a Latin hypercube on the eight design variables of the original problem: wing area, aspect ratio, camber, thickness ratio, maximum camber location, maximum thickness location, root sparcap width and cruise speed. Interdisciplinary compatibility is enforced by fixed point iteration with simple looping with a relaxation factor. A second sample is created with BCS. BCS was warm-started with a

Latin hypercube sample of 50 points at the system level. In both cases, the total sample size is determined by a limit of 5000 ± 10 aerodynamics function calls.[1] The same Bayesian model type is fit to both the DoE and BCS sample. A sparse Bayesian model with a Gaussian kernel function and a bias term is used. The kernel width parameters are selected by optimizing RMS model fit error of the predictive mean using 20-fold cross-validation. Note that BCS models are fit not only to the design variables listed earlier but also to coupling variables (load and deflection distributions, weight), so the model input dimensions are higher than for the DoE.

Next, the actual optimum is located to provide a reference point for comparing DoE and BCS. A genetic algorithm in Matlab is used for this direct optimization. For each design evaluation, FPI is used to enforce interdiscipline compatibility. Using a GA population size of 100, about 22000 aerodynamics and 17000 structural function calls are required to converge. The GA optimization is performed five times and repeated once with simulated annealing to confirm the optimum.

5.4.2 Results of BCS vs. Latin Hypercube DoE

Using the GA and simulated annealing optimizers, the reference optimum design point, $\mathbf{x}_{\text{opt}}$ is:

[1] The sample size can only be limited approximately, because the number of disciplinary iterations to converge each FPI or BCS iteration cannot be accurately predicted beforehand.

aspect ratio:	11.6
thickness/chord:	0.0885
camber/chord:	0.0207
chordwise maximum thickness location:	0.312
chordwise maximum camber location:	0.355
normalized leading edge radius:	0.810
root sparcap thickness:	0.0619 ft
cruise speed:	26.6 ft/s

The optimum response is $L/D_{\text{opt}} = 28.1$, and two of three constraints are active: maximum deflection and minimum sinkrate.

How certain is the designer's knowledge of L/D near $\mathbf{x}_{\text{opt}}$ based on the DoE and BCS samples? The Bayesian models fit to the two samples each give a predictive distribution at $\mathbf{x}_{\text{opt}}$ shown in Table 3. The $\delta_{95\%}$ characterizes a 95% Bayesian confidence interval above and below L/D_{opt} predicted at $\mathbf{x}_{\text{opt}}$, as shown earlier in Figure 28. These results show that the BCS-based model gives more certain predictions at the GA optimum.

Table 3: Comparison of DoE and BCS-based model predictions at GA optimum

	DoE	BCS
Aerodynamics function calls	5000	5000
Best feasible L/D value	25.47	26.64
Predicted L/D mean at GA optimum	22.10	27.42
Predicted L/D st.dev. at GA optimum	9.44	0.027
Bayesian confidence interval $\delta_{95\%}$	**11.03**	**0.93**

The reason for this higher confidence can be seen by from the sample. Figure 40 shows the sample points in the system-level design space of the BCS problem. Note that the figure shows only three of the twelve system design variables. The GA optimum

point is shown as a red asterisk, and the BCS sample points are clustered near it. Figure 41 shows these points with the objective function $-L/D_{\max}$ on the vertical axis. The points are concentrated in a band that includes the optimum.

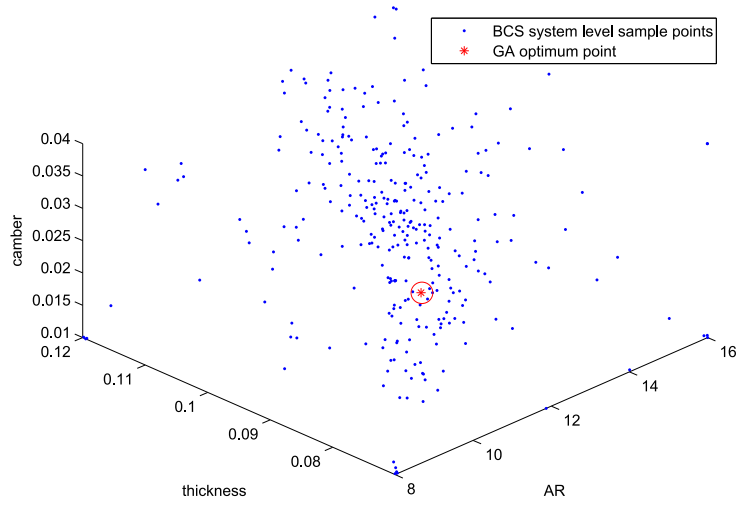

Figure 40: System sample points

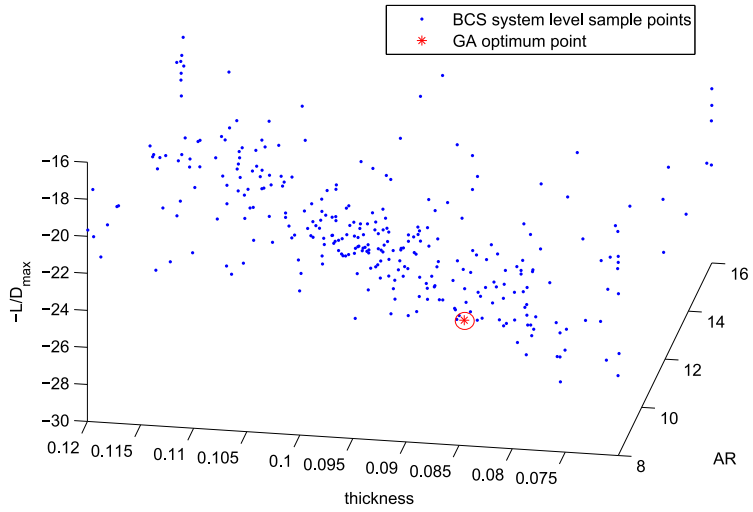

Figure 41: Objective function, $-L/D_{\max}$

The band also tends to lie in regions interdisciplinary compatibility, as can be seen in Figures 42 and 43. These show points plotted for sum of squares of target discrepancy metrics J_k in each discipline — basically, an aggregate measure of compatibility in aerodynamics and structures, respectively. The points cluster in regions of low target discrepancy (or high compatibility).

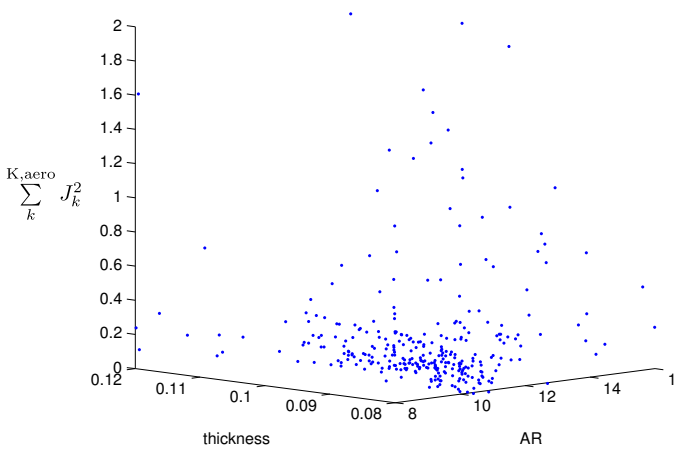

Figure 42: Aggregated aerodynamics target match criteria, $\sum_{k}^{\text{K,aero}} J_k^2$

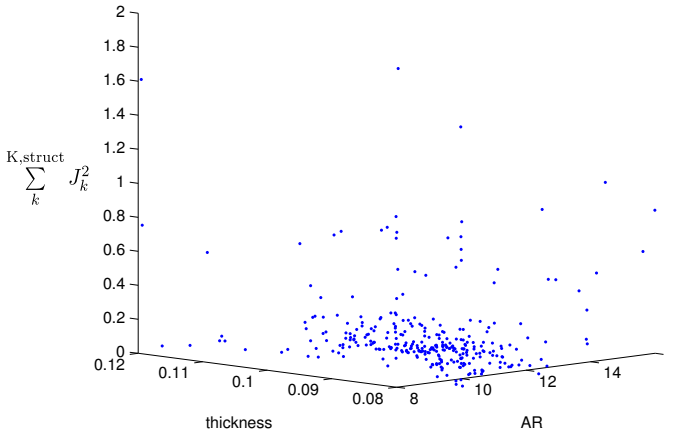

Figure 43: Aggregated structures target match criteria, $\sum_{k}^{\text{K,struct}} J_k^2$

The mechanism for this accumulation of compatible and favorable design points can be shown by graphing the BCS system-level sampling criterion, repeated from Eq. (7):

$$Y_{\text{sampling}} = -\log E\left[I\left(F(\mathbf{Z})\right)\right] - \sum_{k} \log P\left(J_k^2\left(\mathbf{Z}\right) \leq \epsilon\right)$$

where $\epsilon = 10^{-2}$ and i represents each discipline. Figure 44 shows Y_{sampling} after a maximum of 2000 aerodynamics or structures function calls, plotted against two of the design variables, with all other variables set to the GA optimum value. Figure 45

shows the same surface after a maximum of 5000 disciplinary function calls. Three factors account for the differences in the two plots. First, the current best objective value used to compute $E\left[I\left(F(\mathbf{Z})\right)\right]$ has changed. Second, dense sampling near this 2-dimensional slice of the design space has reduced the predictive variance, resulting in more sharply-resolved features. Third, there is higher certainty of the compatibility in the region. This can be shown by decomposing the sampling criterion and plotting only $\prod_i P\left(J_i\left(\mathbf{Z}\right) \leq \epsilon\right)$ versus aspect ratio and thickness, setting all other design variables to the GA optimum values. See Figure 46. The optimum design falls in the region with high predicted probability of compatibility. The scale of probability masses shown in the figure are very small because of a small value of ϵ.

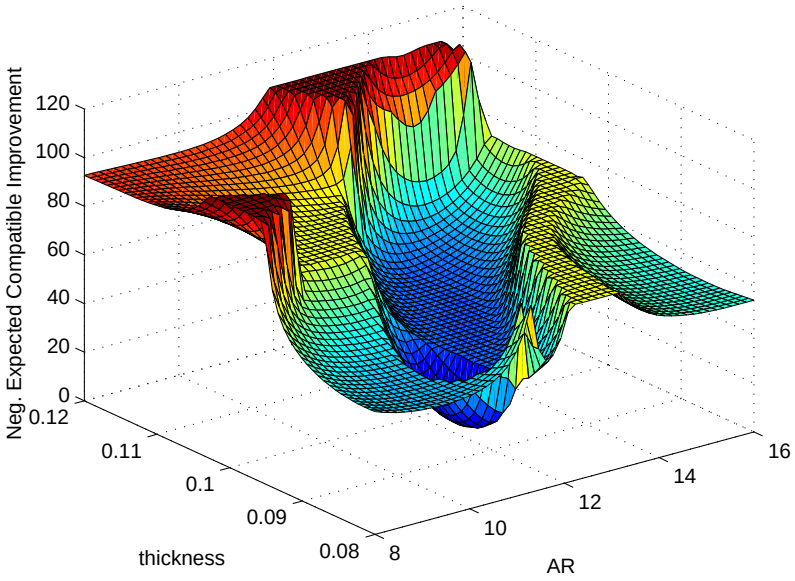

Figure 44: System sampling criterion after 2000 disciplinary function calls

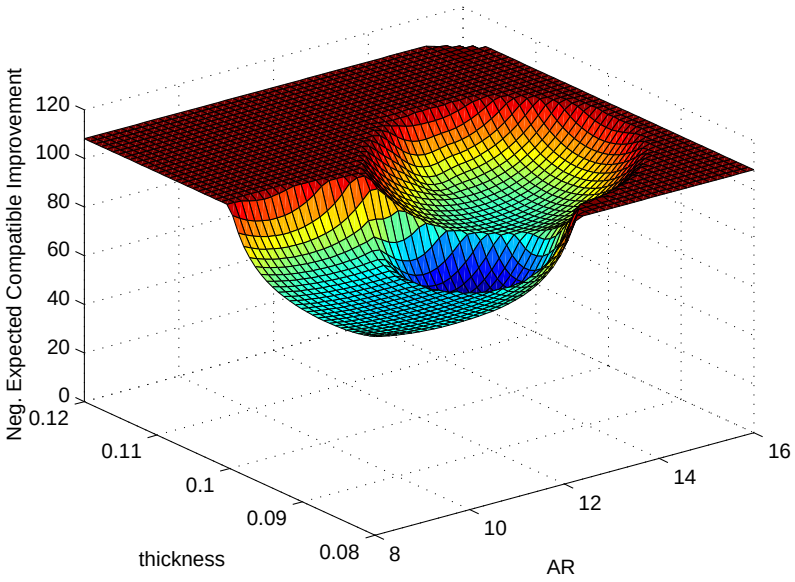

Figure 45: System sampling criterion after 5000 disciplinary function calls

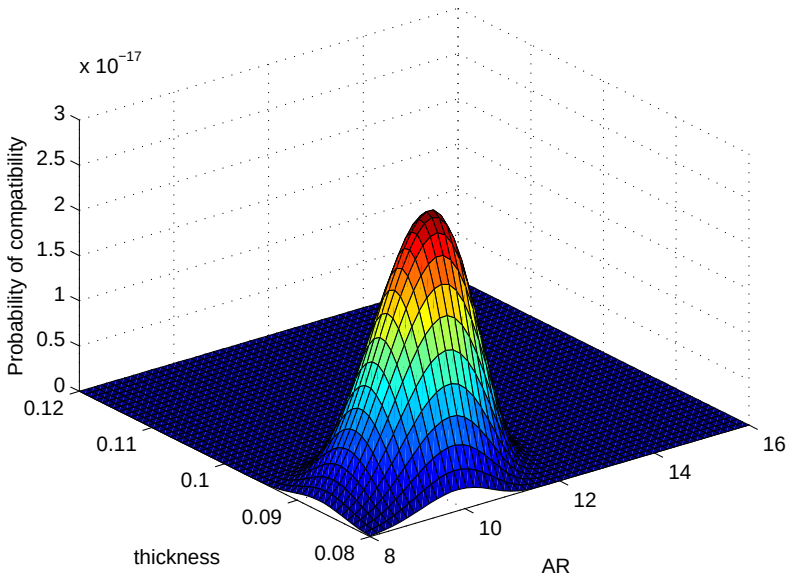

Figure 46: Interdisciplinary compatibility after 5000 disciplinary function calls

At this point, the designer can rapidly view plots like Figures 44 through 46 for different sections of the design space. When reviewing the probabilities of compatibility and expectations of improvement, it is important to remember that these are all based on subjective probability. The plots are only believable insofar as the designer accepts the assumptions of the Bayesian model and choice of basis functions. If the designer believes (based on experience or physical intuition) that models fit the underlying physical phenomena, then these plots can be used to constrict the design space around a small region. For example, in this case, the mostly linear physical equations make it unlikely that the objective function is highly multimodal or contains hidden optima. So, it would be reasonable for the designer to trust the Bayesian models and reject the plateau regions in the plot of expected compatible improvement in Figure 45. A smaller design space can then be used for other tasks like direct optimization. In the worst case, the designer has low confidence that the calculated expected improvement or probability of compatibility correspond to reality. BCS still provides a feasible, compatible, current best solution. The designer can simply accept this

design as the best that can be done with a limited analysis budget. He/she could use wing design and move on to the next design stage, adding detailed variables such as control surfaces, nacelles, tip shapes and root fillets.

At this stage, the second hypothesis and prediction have been addressed for a single test case. While this example cannot *prove* general statements about BCS performance, it supports the claim the BCS concentrates sample points by favorable objective values and interdisciplinary compatibility. Next, the third research question/hypothesis/prediction is examined to characterize when BCS works well. There are perhaps an unlimited number of ways by which BCS may not work well, so the following section focuses on one of the most important factors that could affect its performance.

5.5 BCS Performance and the Effect of Coupling Bandwidth

The number of coupling variables compared to design variables can be expected to affect the performance of BCS. The major advantage of the bi-level decomposition in the original collaborative optimization (CO) and BCS is the avoidance of fixed point iteration (FPI). Recall that the coupling variables in the original MDO problem become system level variables in BCS. BCS trades the expense of FPI in exchange for additional system-level variables, which increases the dimension of the design space. So, BCS would tend to show less of an advantage compared to direct FPI problem as the system level design space becomes large, and this tendency is magnified if the FPI problem is relatively simple and converges easily. The glider problem was modified to artificially vary the coupling bandwidth. The predictions from Hypothesis 3 state that a performance measure of BCS is expected to drop (compared to a baseline) as bandwidth increases. The tests in this section support the conclusion that BCS performance does degrade with coupling bandwidth, subject to the limitations of a single test case.

For this study, it is helpful — although not logically required — to observe some baseline method as the coupling bandwidth varies. There are many possible benchmark methods, but pure EI sampling with FPI is used because it is simple and an obvious default. Note that searching for good objective values while directly enforcing interdisciplinary compatibility with FPI for each individual design is also called a multidisciplinary feasible (MDF) architecture in the MDO literature. The the point of comparing BCS with EI/FPI is not to gauge BCS performance compared to a rival method but to track changes in relative performance as coupling bandwidth increases.

The relative performance of the benchmark EI/FPI is heavily dependent on internal tuning parameters, such as relaxation factors, variable ranges, and convergence tolerances. Slight changes in relaxation can increase the number of function calls per design by an order of magnitude. Unfortunately, it is extremely difficult to select credible, neutral, unbiased settings, and the tuning parameters must be set arbitrarily set to something. Furthermore, some tuning of EI/FPI is unavoidable simply to make the algorithm stable enough for hundreds of design iterations. Therefore, the methodological strategy is to simply choose the other extreme: to explicitly tune the benchmark and not even try to make a neutral performance comparison. Thus, the EI/FPI algorithm thus serves an optional reference for the effect of physical complexity as coupling variables increase. This masking effect is quite imperfect, but it at least addresses the issue. If BCS performs much worse than EI/FPI as coupling bandwidth increases, there is increased confidence that this bandwidth affects performance by a system level curse of dimensionality. If the EI/FPI method fails, the conclusion can still be valid, though it would not be as strong.

The measures of performance are the $\delta_{95\%}$ and $\epsilon_{95\%}$ defined in the previous chapter. They are both evaluated at the optimum. Recall that $\epsilon_{95\%}$ is a metric for accuracy

of the prediction of compatibility, and it is defined by Equation 70

$$P_{\text{compatibility}}(\epsilon_{95\%}) \approx \prod_{k}^{K} P\left(-\epsilon_{95\%} < J_k(\mathbf{Z}) < \epsilon_{95\%}\right) = 0.95 \tag{70}$$

where K is the number of coupling variables, J_k is the target discrepancy between disciplinary subspace and system target, and $\mathbf{Z}$ is a system design. $\epsilon_{95\%}$ is only applied to BCS, since target discrepancies are not computed in the EI/FPI reference method.

The actual optimum of the glider problem is required to compute both metrics, $\delta_{95\%}$ and $\epsilon_{95\%}$. Optimization was tried in Matlab using the *fminsearch* (gradient-free simplex-based optimizer), *fmincon* (trust region reflective), and a genetic algorithm. The genetic algorithm with default settings proved to be too expensive. Using an Intel i7 2.93 GHz desktop computer, the genetic algorithm would not converge after wall-clock times of several days and $O(10^5)$ aerodynamic function calls. Instead of using a global optimizer, the fminsearch was used with random initial points. Because of the chance of finding only local optima, the search was repeated five times, with typical cost of $O(10^4)$ aerodynamic function calls. In certain trials, even this was not enough, and BCS located a better objective value than the optimizers. In these cases, the best result from BCS was used as an initial point for Matlab's *fmincon* optimizer and used to confirm the optimum. Although there is no guarantee of a global optimum, this was considered sufficient for the present purpose, especially since the Bayesian methods would provide some additional assurance that a global optimum is lurking in some unexplored region of the design space.

While running the two adaptive samplers, all Bayesian inference was done using sparse Bayesian models. Each of these experiments was warm-started with 200 aerodynamics function calls from a Latin hypercube design for BCS and approximately 200 random function calls for EI/FPI. Random points were used for EI/FPI because it was difficult to predict how many system designs are required to produce 200 aerodynamics function calls, and it is therefore difficult to generate a Latin hypercube with the correct guess for the required number of system design points. The 200

function call warm-start sample size is arbitrary and the experiments' dependence on this sample size is explored in a later section. If the warm-start is set to a very small number, BCS tends to sample the corners of the design space — it essentially performs an exhaustive full factorial experimental design at low and high extremes of each variable. In informal experience, it was found that 200 warm-start points provided sufficient assurance that BCS would have enough information to sample in the interior of the space and not become stuck sampling corners. Experiments later in this chapter show that this was indeed a reasonable choice.

A number of parameters in the sparse Bayesian models had to be set arbitrarily. These tuning parameters can certainly affect the performance of BCS or EI/FPI, yet they have to be set to some value. The best that could be done was to keep those values constant throughout the tests, recalling that the main purpose of these tests is not a contest of general performance. The basis functions were set to Laplace kernels as well as parametric functions: Z, Z^2, Z^3, $\sin(\pi Z)$, $\sin(2\pi Z)$, and $\sin(4\pi Z)$. The Laplace kernel widths were set to 0.3. The width was determined by performing 10-fold cross validation on the first warm-start sample and finding the width that minimized error of the mean prediction. The width was then approximately halved to account for the higher resolution required for more tightly clustered sample that the adaptive sampling schemes would create. These settings were kept constant for three tests each of BCS and EI/FPI. Note that this is different from other BCS tests in this thesis, where the kernel widths are automatically tuned by optimizing RMS error by k-fold cross-validation. In those experiments, the "hands-off" performance and elimination of arbitrary parameters are important to show BCS performance with credibility. In this case, however, the width is frozen to try to isolate the effect of coupling bandwidth. In other words, if BCS performance suffers in these present tests, we want to know if it is due to coupling bandwidth, rather than incidental differences in the automatic tuning of Bayesian models.

Note that the Bayesian models for BCS and EI/FPI have different system design sample sizes and design space dimensions, although the number of aerodynamics function calls were kept approximately constant. The EI/FPI input variables include only the true design variables, while the coupling variables were used only within the FPI process. Each design evaluation required several fixed point iterations, so the number of recorded outputs is substantially less than the number of aerodynamic function calls. On the other hand, BCS uses an augmented design space that includes both the true design variables as well as coupling variables. The purely local discipline variables (such as structural spar cap width) are considered only in the disciplinary sub-optimization, and are thus removed from the system design space. Also, BCS records system level responses like the objective function and compatibility metrics only for each system iteration. Finally, BCS does not enforce compatibility strictly at each outer/system iteration, but gradually moves sampling to regions of probable or approximate compatibility. In contrast EI/FPI attempts to enforce compatibility at every design.

Since models fit to BCS and EI/FPI are based on different sample sizes and input dimensions, is this not comparing "apples to oranges?" Is the comparison of $\delta_{95\%}$ for different input dimensions meaningful? First, philosophically, recall that designer is mainly concerned about learning about favorable regions or having certainty near the optimum. Regardless of whether compatibility closure is enforced at every data point or how big the input space is, the designer simply wants a confident approximation near the optimum. Secondly, recall that we are not actually comparing apples to oranges, but using one to mask certain effects while studying the other. We are not trying to compare BCS and EI/FPI performance as such, but using EI/FPI performance to benchmark physical complexity while studying the effect of coupling bandwidth in BCS. Therefore, it is assumed that the difference in sample sizes and input dimensions does not confound the study of Hypothesis and Prediction 3. The

final uncertainty near the compatible, optimum region is what matters to the designer.

In the tests, the previous glider example was reused with minor alterations. The cruise velocity was removed from the design space and set to a constant 25 ft/s for the sake of computational cost.

The coupling bandwidth was varied by increasing the resolution of wing deformation. The deformation of the wing was expressed as discontinuous, polyhedral breaks instead of a continuous curve. This was done so that the curved wing could be approximated in the vortex lattice code by a polyhedral wing with multiple, flat wing panels. The deformation coupling variable is simply the transverse displacement at these spanwise discontinuities. The number of polyhedral breaks was used to increase coupling bandwidth. It is also possible to increase the coupling bandwidth by increasing resolution of aerodynamic loads passed from the aerodynamics to structures discipline. In this example, the spanwise distribution of transverse loads were parametrized as a third order polynomial. Increasing the order of this polynomial could increase the coupling bandwidth. However, such coupling variables were not added in order to control changes in physical complexity while changing bandwidth.

In this example, the number of aerodynamics function calls is used as the cost metric because the aerodynamics simulation is more costly than the structural code, which is but a simple integration of the Euler beam equation. BCS and EI/FPI were run to 2000 aerodynamics iterations and the performance metrics were recorded at 500, 1000, and 2000 iterations. This is a smaller number than the 5000 iterations used in Hypothesis 2, which was used to answer a more basic question of whether BCS works at all.

In all the following examples, the Bayesian sampling criterion is composed of the simple EI and target matching probabilities (recall that the previous example used a

J_k^2 term that was deemed an unnecessary complication).

$$-\log E\left[I\left(F(\mathbf{Z})\right)\right] - \sum_k \log P\left(\epsilon < J_k\left(\mathbf{Z}\right) < \epsilon\right) \tag{71}$$

5.5.1 Coupling Bandwidth Test Results

The results of this analysis are shown in Figure 47 as well as Tables 4 through 9. The BCS trials show that the uncertainty at the optimum of both the objective and compatibility ($\delta_{95\%}$ and $\epsilon_{95\%}$) increases (uncertainty increases) with number of coupling variables — if comparing for similar numbers of aerodynamics function calls. This provides some limited support for Hypothesis 3: as coupling bandwidth increases, BCS accuracy at the optimum degrades due to the increased dimensions of the system level search space. The tests cannot rule out other confounding factors, but they are strongly suggestive.

On the other hand, the EI/FPI trials do not show a clear trend and are not so useful even as a reference baseline. Detailed examination of the EI/FPI tests show that variation in $\delta_{95\%}$ is strongly influenced by the internal FPI solver performance. The cost in disciplinary iterations of the simple looping FPI algorithm with adaptive relaxation rate was highly variable, so that some designs required 4 iterations to converge coupling variables while others required as many as 600. The performance of EI/FPI was strongly influenced by the stability and efficiency of FPI, since a single design FPI could consume a large share of the entire computational budget. As a consequence, the performance is highly sensitive to the initial sample. For this reason, the EI/FPI was not a very useful benchmark. Nonetheless, the BCS trials alone show some trends that support Hypothesis 3: performance measured by $\delta_{95\%}$ and $\epsilon_{95\%}$ show a dependence on coupling bandwidth.

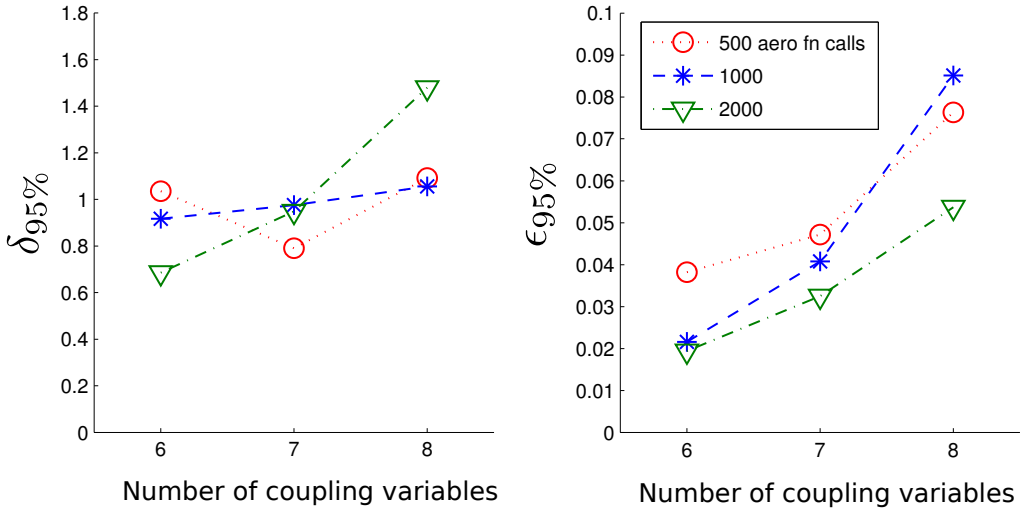

Figure 47: BCS performance with varying coupling bandwidth for 500, 1000, and 2000 aerodynamics function calls

Table 4: BCS for 6 coupling variables; optimum = 27.4

Aerodynamics function calls	500	1000	2000
Structures function calls	5179	10103	19844
Best compatible objective value	26.27	26.72	27.21
Predicted objective mean at optimum	26.64	27.03	26.99
Predicted objective st.dev. at optimum	0.161	0.326	0.162
BCS confidence for objective, $\delta_{95\%}$	**1.035**	**0.917**	**0.686**
BCS confidence for compatibility, $\epsilon_{95\%}$	**0.038**	**0.022**	**0.019**

Table 5: EI/FPI for 6 coupling variables; optimum = 27.4

Aerodynamics function calls	505	1003	2006
Structures function calls	505	1003	2006
Best compatible objective value	25.23	25.23	25.23
Predicted objective mean at optimum	25.50	24.59	33.78
Predicted objective st.dev. at optimum	27.42	4.99	3.54
BCS confidence for objective, $\delta_{95\%}$	**54.00**	**18.05**	**12.40**

Table 6: BCS for 7 coupling variables; optimum = 27.4

Aerodynamics function calls	500	1000	2000
Structures function calls	5146	10103	19844
Best compatible objective value	26.15	26.73	26.98
Predicted objective mean at optimum	26.93	26.89	26.94
Predicted objective st.dev. at optimum	0.191	0.279	0.294
Confidence for objective at optimum, $\delta_{95\%}$	**0.790**	**0.975**	**0.949**
Confidence for compatibility at optimum, $\epsilon_{95\%}$	**0.0472**	**0.0408**	**0.0325**

Table 7: EI/FPI for 7 coupling variables; optimum = 27.4

Aerodynamics function calls	759	1175	2169
Structures function calls	759	1175	2169
Best compatible objective value	22.30	22.30	22.30
Predicted objective mean at optimum	21.71	21.13	22.78
Predicted objective st.dev. at optimum	142.96	0.92	1.03
BCS confidence for objective, $\delta_{95\%}$	**75.49**	**7.67**	**6.19**

Table 8: BCS for 8 coupling variables; optimum = 27.4

Aerodynamics function calls	500	1000	2000
Structures function calls	5090	9877	19643
Best compatible objective value	26.124	26619	27.101
Predicted objective mean at optimum	26.66	26.79	26.92
Predicted objective st.dev. at optimum	0.208	0.264	0.602
Confidence for objective at optimum, $\delta_{95\%}$	**1.092**	**1.056**	**1.480**
Confidence for compatibility at optimum, $\epsilon_{95\%}$	**0.0763**	**0.0851**	**0.0537**

Table 9: EI/FPI for 8 coupling variables; optimum = 27.4

Aerodynamics function calls	640	1373	2061
Structures function calls	640	1373	2061
Best compatible objective value	24.51	24.51	24.51
Predicted objective mean at optimum	21.63	10.80	33.66
Predicted objective st.dev. at optimum	3.33	17.66	22.70
BCS confidence for objective, $\delta_{95\%}$	**45.50**	**21.20**	**45.97**

The almost patternless results of EI/FPI are due to a highly variable internal FPI algorithm. The first EI/FPI case was tuned for reliable performance, and most designs converged with less than ten disciplinary iterations. The same tuning parameters were used with 7 and 8 coupling variables. In the final case with 8 coupling variables, the number of FPI function calls per individual design ranged from 4 to 230. This led to a small EI/FPI sample of converged designs that was very dependent on the random warm-start sample.

The results also show a peculiarity for BCS for a fixed setting of coupling bandwidth. How can the uncertainty metric $\delta_{95\%}$ increase as the number of samples increases in some cases? See the circled column of points with 7 coupling variables in Figure 48.

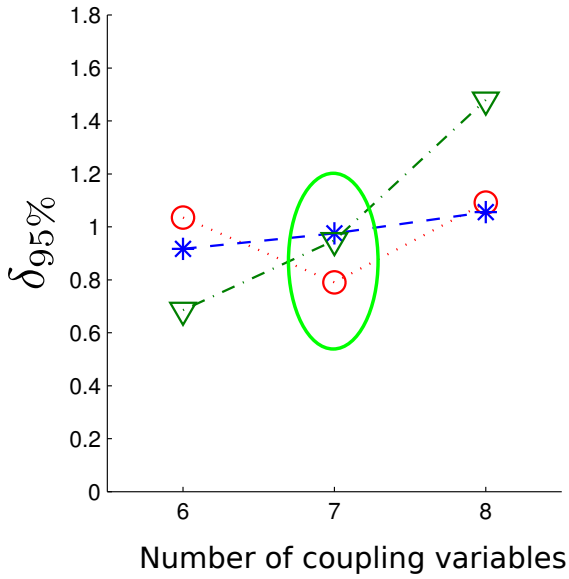

Figure 48: BCS performance with varying coupling bandwidth for 500, 1000, and 2000 aerodynamics function calls

The answer may lie with the sparse Bayesian model. In the regression process, not all of the basis functions are activated due to automatic relevance determination. If a nearby basis function happens to be activated, then the predictive variance at nearby regions tends to be smaller. Similarly, if many of the parametric functions are inactive, then the model has fewer available modes with which to express uncertainty, leading to inconsistent trends in predictions. While this phenomenon is certainly undesirable, the effect is relatively small in these examples – it is part of the cost paid for the speed of sparse regression. The effect would tend to diminish as more sample points and bases are added to the model.

Again, recall that the foregoing is only a single test problem, and perhaps the only way to study this hypothesis about coupling bandwidth with high certainty is to gather data for a large number of engineering problems. Also, because of the dependence on random warm start samples, these experiments should ideally be replicated many times to test the hypothesis very completely. However, the extent of the study is limited by computational cost and the potential impact of its findings for the hypothesis. Despite the very high speed of sparse Bayesian models compared to Kriging,

each BCS trial requires days of wall-clock time to complete. The optimization to find the reference point for performance measure requires closer to a week due to the need for multiple trials with different initial points. Realistically, it is most likely that a small number of practictioners may choose to use the experimental BCS method based on qualitative reasons, such as disciplinary autonomy. It will be the weight of accumulating experience that shows whether coupling bandwidth is the crucial criterion for using BCS. The modest purpose of coupling bandwidth tests with the glider problem is to show that this mechanism is plausible.

5.5.2 Sensitivity to Warm-Start Sample Size

Although it is not a central point, a limited investigation of the size of the warm-start population was also performed. In the above study of coupling bandwidth, a Latin hypercube sample of 200 points was used to initialize BCS. This number was chosen from experience but was still arbitrary. How sensitive is BCS to this initial sample? What are the risks of selecting the "wrong" warm-start size?

Assume that the Bayesian models used in BCS have a form in which predictive variance increases with distance from observed points. This is true of simple Kriging. It is also true of sparse Bayesian kernel models that have parametric basis functions, such as x, x^2 or $\sin(x)$, along with kernel function centered on data points. Again, pure kernel models such as RVM often do not have this desirable quality, because the value of the kernels decays with distance, so the basis functions are incapable of expressing high variance far from sample points.

With such Bayesian models, suppose that an adaptive sampling algorithm is warm-started with a relatively small sample. New points in the interior regions of the design space will tend to have more nearby points and lower variance than at the edges or corners of the design space. This tendency is greatly magnified as design dimensions increase. Consider a hypercube (multidimensional cube) design space for K design

variables, where design ranges are of length 2 for simplicity. For $K = 2$, the ratio of distance from a corner (vertex) to the square center compared to distance of a face(side) center to the square center is $\frac{\sqrt{2}}{1}$. Generally, this ratio of distances is $\sqrt{K}$ and, of course, $\lim_{K\to\infty} \sqrt{K} = \infty$. The corners are far from the center, and this distance increases with hypercube dimensions. By similar argument, distances between corners that do not share an edge of the hypercube also increase as dimensions increase.

With an insufficient warm-start, there is often large uncertainty in the corners of a design space, because they tend to be far from other points. Assume that the adaptive sampling algorithm first places a new sample point in a corner. At higher dimensions, corners are far from one another, so the sampling of one corner has weak influence on the uncertainty at another corner. At each iteration, the sampling algorithm may select the point farthest from all others, and will tend to place new sample points sequentially in corners. The problem is that there are many corners — 2^K of them. The Bayesian sampling method tends to behave like a full factorial design of experiments with two settings (high/low) for each design variable. This tendency continues until there is enough information for patterns to emerge, to regress basis functions to have wide influence, or to activate parametric basis functions such as 1, x, and x^2.

With a sufficient warm-start sample, enough information about the underlying function is detected from the data so that variance is relatively low at the extremities of the design space. Whatever the sampling criterion is, the variance at the extremities must small enough to avoid becoming mired by the corner-sampling phenomenon. The ideal warm-start size is very problem-dependent and model-dependent. For example, if using sparse Bayesian models on a linear function with very little noise, a small sample size will establish the general form of the underlying function and decrease variance at the extremities of the design space.

The consequences of an excessively large warm-start sample are comparatively benign. In the extreme, the adaptive sampling becomes an offline DoE. Simply, there is not much remaining budget with which to exploit accumulating information.

An informal study of warm-start sample size was done with the glider test case. In each trial, the warm-start was done with a Latin hypercube sample created in Matlab. Latin hypercube generation depends on random seeds, so if more computer resources were available, each trial should be replicated many times. However, BCS sensitivity to the warm-start likely depends strongly on the particular problem, so there is limited benefit in very precisely characterizing this sensitivity in a large scale experiment. The results in Table 10 simply show that there can be a dependence on warm-start samples. In the table, warm-start sample size is expressed in terms of the number of initial aerodynamics function calls. The last column with a warm-start of 1000 function calls is the degenerate case where the entire budget is used for a Latin hypercube DoE. The "best objective value' field is empty for this case because no compatible designs were found. Figure 49 shows the same information in a graph.

As one might expect, BCS suffers at small warm start values because it must establish general patterns in the underlying function; in the opposite extreme, performance suffers with excessively large warm-starts that are too similar to complete off-line sampling.

Table 10: Comparison of BCS Performance with Variable Warm-Start Sample Size

Warm-start sample size	10	100	200	500	1000
Final aero fn calls	1000	1000	1000	1000	1000
Final struct fn calls	9329	9992	10103	19844	10764
Best objective value	26.85	27.30	26.72	26.01	—
Pred mean at opt	26.97	27.03	27.02	26.06	24.81
Pred stdev at opt	0.98	0.253	0.286	0.530	0.024
$\delta_{95\%}$	**1.98**	**0.60**	**0.67**	**2.02**	**2.43**
$\epsilon_{95\%}$	**0.0727**	**0.0051**	**0.0074**	**0.0517**	**0.0860**

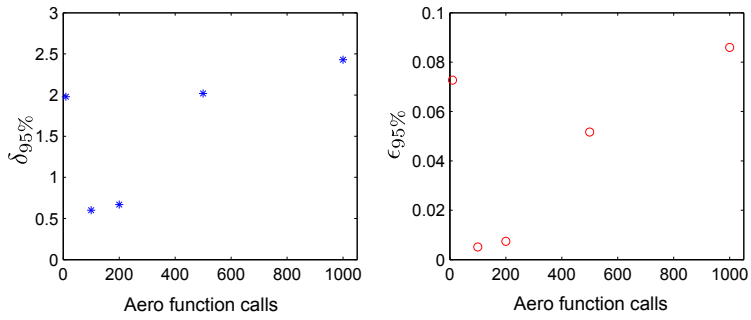

Figure 49: Comparison of BCS Performance with Variable Warm-Start Sample Size

What is the designer to do? As an informal rule of thumb, one might devote 10-30% of the total analysis budget to the warm-start sample, as was done successfully in the glider case and the following engine nacelle case. But the total budget will vary for different engineering projects. The best advice may be to begin with a modest warm-start, fit the Bayesian models, and check uncertainties at design space extremities to see if there is excessive uncertainty. If so, then more random warm-start points may be added. The same can be done if BCS initially exhibits the corner-sampling phenomenon. Although Bayesian adaptive sampling and BCS are sensitive to warm-start sample size, there is relatively low practical risk, since more random warm-start points can be added if a difficulty arises.

5.5.3 Replication Error

The warm-start tests show that the initial sample before adaptation plays a significant role in the performance of BCS. In all trials, a Latin hypercube sample was used for the warm-start. Because such samples are generated from a random seed (in Matlab), there is variation in replicated trials. In addition to this variation, there is also replication error due to the stochastic optimizers (genetic algorithms) used to optimize the Bayesian sampling criterion at every system iteration.

To estimate this error, the BCS case with six coupling variables and 200 initial aerodynamic warm-start function calls was repeated eight times. Each test requires several days of wall-clock time, so the sample size is relatively small but sufficient

to show the general idea. Results are shown in Table 11. The results show that

Table 11: Sample statistics for performance of 8 replicated BCS trials, 6 coupling variables, 200 point warm-start

	$\delta_{95\%}$	$\epsilon_{95\%}$
sample mean	0.864	0.0356
sample st.dev.	0.276	0.0251

there is significant variation across trials. This is not surprising because a 200 point warm-start sample likely covers a very small subspace of the 13-dimensional design space. Randomly generated DoEs of that size will yield very different Bayesian models and predictive distributions to begin the BCS adaptive sampling. This is likely an unavoidable difficulty in performing BCS with small sample sizes. However, the replication error would tend to be alleviated as the BCS sample size increases.

5.5.4 Conclusions

The low fidelity glider test problem was used to explore performance and theoretical issues of BCS. Despite the use of fast potential flow and beam theory models, the BCS tests were still expensive with turnaround times on the order of days. But the simple aero-structural case allowed variation of problem to test predictions of Hypothesis 2 and 3. There is preliminary support for the claims that: 1) BCS concentrates points based on a probably favorable objective and probable compatibility, and 2) coupling bandwidth is a key factor that affects BCS performance.

This investigation of specific mechanisms of BCS in a realistic problem may guide practitioners who consider BCS or similar Bayesian approaches for their problem. The next chapter turns to Research Question 4, which is even more explicitly concerned with practical tradecraft: can BCS be implemented for a high fidelity code like CFD in a realistic and more physically complex problem?

CHAPTER VI

PRACTICAL BCS IN HIGH FIDELITY ANALYSIS

Research Question 4 asks whether BCS can be used in a practical, high fidelity engineering problem. While less interesting from a purely theoretical perspective, a large part of the research effort is devoted to this question because of its importance to the designer's trade. The question is deliberately open-ended, without formal hypotheses or prediction testing as in the previous chapter. The 'methodology' is simply to apply BCS to a high fidelity engineering problem and discover pitfalls and lessons.

The high fidelity example is the aero-propulsion design of an isolated engine nacelle. This test case is selected mainly because it contains many of the essential characteristics of a boundary layer ingesting aircraft aero-propulsion design while being computationally feasible for this thesis. A meridional view (radial-axial plane) of the engine is shown with Mach numbers in Figure 50. The CFD costs of such

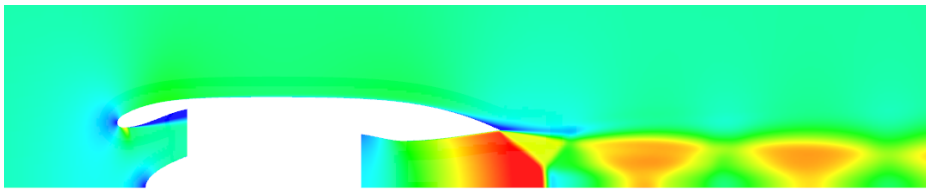

Figure 50: Powered engine nacelle

simulations are high because of high flow gradients and viscous and compressible phenomena in the inlet and nozzle. However, the engine is axisymmetric, CFD costs are substantially lower than, for example, an entire aircraft. This allowed a full BCS demonstration with a thousand CFD function calls as well as several thousand more calls while debugging and solving BCS implementation problems.

In addition to the engine nacelle example, preliminary studies were done with the Boeing/NASA N2B hybrid wing-body (HWB) with boundary layer ingestion (BLI).

This example was discussed in the introductory chapter because it showcases both the need for high-fidelity analysis in early design stages as well as the challenge of interdisciplinary compatibility. As will be shown, the computational requirements for demonstrating BCS with BLI HWB proved to be beyond the budget of this thesis, but the example is described in this chapter to report lessons from the preliminary efforts and to provide a record for future researchers.

Figure 51 shows an initial baseline design for the BLI HWB aircraft concept.

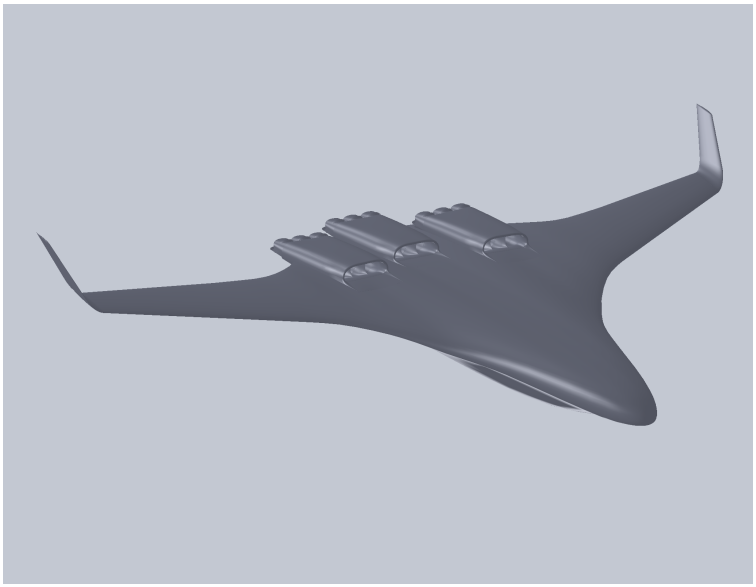

Figure 51: Boeing/NASA N2B hybrid wing-body

This chapter first describes the engine nacelle problem and its BCS implementation and results. Then, it describes the challenges of the BLI HWB design problem.

6.1 Powered Engine Nacelle

The main high fidelity BCS demonstration was done with a powered nacelle example. More specifically, it focused on aero-propulsion design of a nacelle, inlet, nozzle, and the propulsion cycle of a turbojet engine. This example was chosen primarily because the coupled design problem has similarities with the BLI aircraft example, yet is computationally inexpensive. CFD was used for the external flow as well as

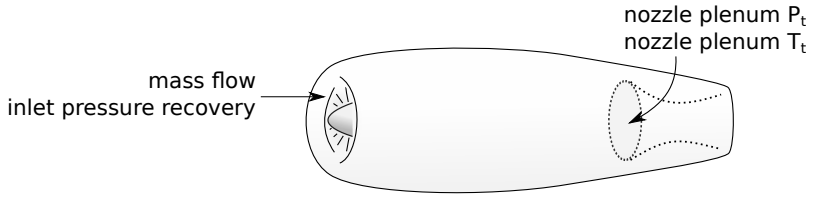

Figure 52: Aero-propulsion coupling variables for engine nacelle

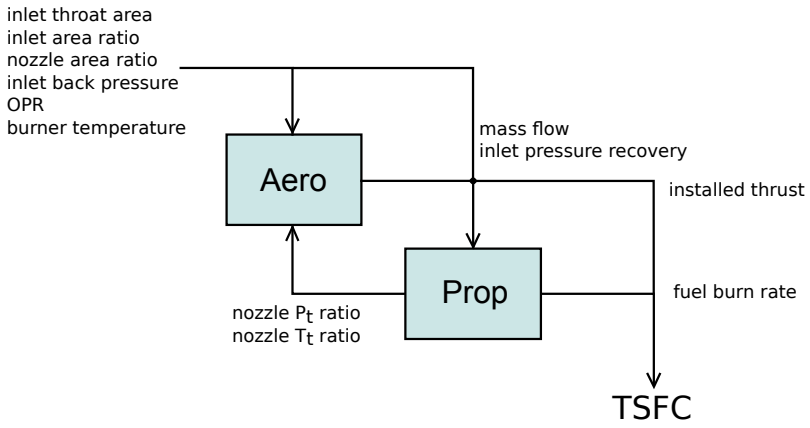

Figure 53: DSM for engine nacelle

inlet and nozzle duct flow. A numerical propulsion cycle code was used to model the internal engine components, from compressor inlet to turbine exit. There is an interdisciplinary coupling in which the nacelle and inlet aerodynamics provide a mass flow and total pressure recovery to the engine's compressor, and the propulsion cycle produces inlet back pressure and nozzle conditions that determine the exhaust plume in the aerodynamics discipline.

The plume may not affect the nacelle aerodynamics as significantly as it affects BLI HWB airframe aerodynamics, so the two-way coupling is not as strong. However, if one views the entire HWB as an "inlet" for its engines, it is apparent that the design problems are not so different. The design structure matrix (DSM) is shown in Figure 53. The nacelle problem can be considered as a surrogate problem for the more expensive BLI HWB problem. Despite the similarities of the nacelle and HWB design problems, the nacelle example is computationally much cheaper due to axisymmetry. The CFD simulation is essentially two dimensional, considering only

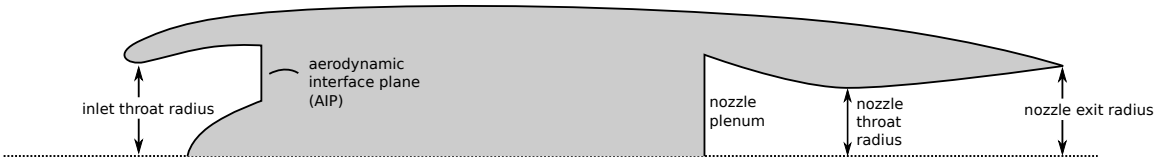

Figure 54: Meridional section view of engine/nacelle

axial and radial flow on a 2° circumferential slice, with periodic boundary conditions in the circumferential direction. On the other hand, this cost saving is offset by a much more detailed study of the internal inlet and nozzle flow of the engine.

A subsonic turbojet example was chosen for simplicity. Turbojets are not commonly used in modern aircraft because turbofans tend to have higher propulsive efficiencies, especially in subsonic applications. However, a turbojet was used to demonstrate BCS and the essential features of coupled aero-propulsion design while avoiding the complexities of a turbofan. These complexities include multiple exhaust streams and the computational cost of modeling the shear flow in the mixing core and bypass streams. A turbofan would increase complexity without adding clarity to the BCS demonstration. Of course, the sampling method can later be generalized to a turbofan or other coupled aero-propulsion problems.

The simple, baseline turbojet design bears some similarities to the Pratt & Whitney J-57 turbojet engine, which was used on the Boeing 707 subsonic transport and military aircraft such as the B-52 bomber. The propulsion cycle model assumes technology levels of the 1950-60s. It is a single spool engine with baseline mass flow of around 200 lbm/s, thrust of around 10,000 lbf, thrust-specific fuel consumption (TSFC) of around 1.2 lbm/lbf-hr. Design ranges include a maximum burner temperature (T_4) of 2400°R and maximum overall pressure ratio (OPR) of 12.0.

Figure 54 shows the key regions and reference dimensions of the engine nacelle. Table 12 shows design variables, coupling variables, and their ranges. The major surfaces are created with cubic splines and Laplacian smoothing in a Python script and with Chimera Grid Tools. The parameterization of nacelle, inlet, and nozzle is

Table 12: Engine/nacelle design variables and ranges

Design Variable	Lower Bound	Upper Bound
inlet throat area	1.0	1.4
inlet area ratio	0.9	1.2
nozzle area ratio	1.5	2.5
mass flow	150 lbm/s	250 lbm/s
inlet total pressure recovery	0.9	1.0
inlet back pressure	1.0	1.5
overall pressure ratio	6.0	12.0
burner static temperature	1800°R	2400°R
nozzle total pressure	2.0	5.0
nozzle total temperature	2.0	5.0

done with an emphasis on physically meaningful design variables. Many of the design variables are normalized or non-dimensionalized by reference quantities calculated by one-dimensional flow models. Total pressures and temperatures are normalized by free-stream static values. The variable called "inlet throat area" is actually the ratio of the throat area with the streamtube capture area for the design Mach number and target mass flow. The inlet area ratio is the ratio of the aerodynamic interface plane (AIP) or to the throat area. The nozzle area ratio is the exit area divided by the critical (choke) throat area given nozzle plenum total conditions and target mass flow. It was found that the critical area in RANS CFD is reliably predicted by 0.735 of the value from a one-dimensional, isentropic flow model using total conditions at the nozzle plenum.

$$\text{design nozzle area ratio} = \frac{A_{\text{exit}}}{A^*} = \frac{\text{nozzle exit area}}{\text{nozzle throat area}} \tag{72}$$

$$\text{nozzle throat area} = A^* = \frac{0.735\ (\text{mass flow target})}{\sqrt{\gamma\, P_{t,\text{nozzle plenum}}\, \rho_{t,\text{nozzle plenum}} \left(\frac{2}{\gamma+1}\right)^{((\gamma+1)/(\gamma-1))}}} \tag{73}$$

Here, γ is the ratio of specific heats, $P_{t,\text{nozzle plenum}}$ is the stagnation pressure at the nozzle plenum, and $\rho_{t,\text{nozzle plenum}}$ is the total density. This simple test case assumes a calorically perfect gas, and γ is assumed to be a constant 1.4. A more realistic simulation would likely have a lower value due to high temperature effects. The

working fluid is also assumed to be standard air, with no adjustment for fuel mass and combustion products.

For realism, a inlet cone/spinner is added with an arbitrary maximum radius of 40% of the compressor (or AIP) tip radius. A circular nacelle lip shape is assumed, allowing for ingestion of a constricting streamtube at starting conditions. To ensure realistic compressor performance, a constraint penalty was applied to Mach numbers at the AIP outside of the range 0.4 - 0.55. Other physical and geometric assumptions are listed in Table 13.

Table 13: Miscellaneous nacelle design settings

Design Settings	Value
cruise Mach number, M_∞	= 0.8
T_∞	= 411.69°R
ρ_∞	= 8.89528e-4 slug/ft^3
nacelle length	= 10.0 ft
inlet length	= 2.0 ft
nozzle length	= 3.0 ft
spinner cone length	= 0.7 inlet length
spinner radius	= 0.4 compressor tip radius
nacelle wall thickness	= 0.15 compressor tip radius
nacelle lip radius	= 0.5 nacelle wall thickness
nacelle plenum area	= 1.3 nacelle throat area

6.1.1 Note on Nozzle Design and Analysis

The nozzle duct geometry was originally not intended to be designed. The flow was originally intended to be modeled in the propulsion cycle code with empirical nozzle loss models to yield the thermodynamic conditions at the nozzle exit plane. These were then to be modeled in CFD as a disc-shaped boundary condition patch that produced an exhaust plume with total temperature and total pressure specified by coupling information from the propulsion discipline. (As a side note, the boundary condition for an engine nozzle outflow is called a CFD *inflow*; the flow enters the CFD domain, even though it *exits* the engine in reality.) It was found that this inflow patch

can cause severe convergence problems, especially during early time-stepping from an initial uniform flowfield. The sluggish outer shear flow from the nacelle boundary layer mixes with the very energetic exhaust plume. This can cause numerical as well as physical instabilities. The velocity shear may lead to physical Kelvin-Helmholtz instabilities, particularly if the uniform exhaust boundary conditions are imposed, with no radial variation of axial velocity to alleviate the sudden discontinuity. The high flowfield gradients may also cause numerical instabilities. To counter the latter, higher order numerical schemes or dense meshes can be used in that region. However, these tend to strongly limit the time step for stable Courant-Friedrichs-Lewy (CFL) numbers. Secondly, even though the turbojet engine is intended for subsonic cruise speeds, it is not uncommon for nozzle flow on such engines to be supersonic. Sampling within the design variable ranges can cause the plume to switch from subsonic to supersonic flow, and the Mach number may also fluctuate about unity as a single CFD solution evolves. The problem is that certain CFD inflow boundary conditions are stable and accurate according to the Mach regime. For example, one common inflow boundary setting uses characteristic boundary conditions to enforce total temperature and pressure while extrapolating mass flow and other properties from the rest of the flowfield. Such a method only works accurately for subsonic flow, since characteristic information is carried upstream from the plume back to the nozzle boundary. On the other hand, supersonic inflow boundary conditions might require specification of many more properties since flow information is carried only downstream. Setting a nozzle boundary condition in a regime where Mach number may vary substantially leads to instabilities and crashes in the BCS sampling method.

To avoid these difficulties, an artificial nozzle chamber and converging-diverging nozzle is created. The nozzle flow begins at a nozzle plenum where the flow is known to be subsonic. The subsonic flow constricts to a sonic throat and expands to become supersonic. In the majority of design cases, the nozzle exit flow is supersonic and

exhibits Mach discs in the plume (analogous to shock diamonds in two dimensional flow). These are shown in Figure 55. This may seem strange to engineeers accustomed to modern high bypass ratio turbofans for subsonic applications, which typically have only a converging nozzle for slightly underexpanded subsonic nozzle exit flow. However, the supersonic exhaust is reasonable for the simple subsonic turbojets, which tend to have much higher nozzle pressure ratios.

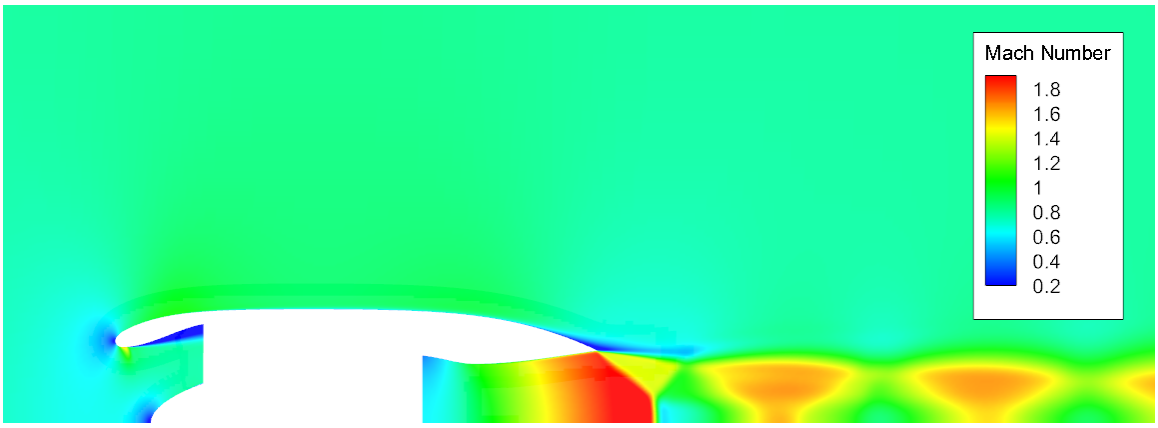

Figure 55: Turbojet with converging-diverging nozzle and shock diamonds (near-field view)

Having described the basic engine nacelle design problem, its implementation in BCS is described in the following sections. As in the glider problem, the steps are described in terms of the four items: bi-level formulation, setup of analysis codes, initialization, and adaptive sampling. Then, BCS results are shown and discussed.

6.1.2 Bi-Level Decomposition for BCS

The first step of bi-level decomposition is highlighted in Figure 56.

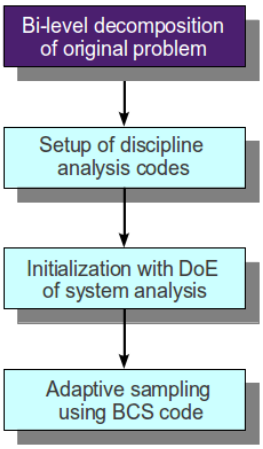

Figure 56: Bi-level decomposition is the first step in implementing BCS

As before, the original problem is decomposed into a bi-level BCS problem, shown in Figure 57.

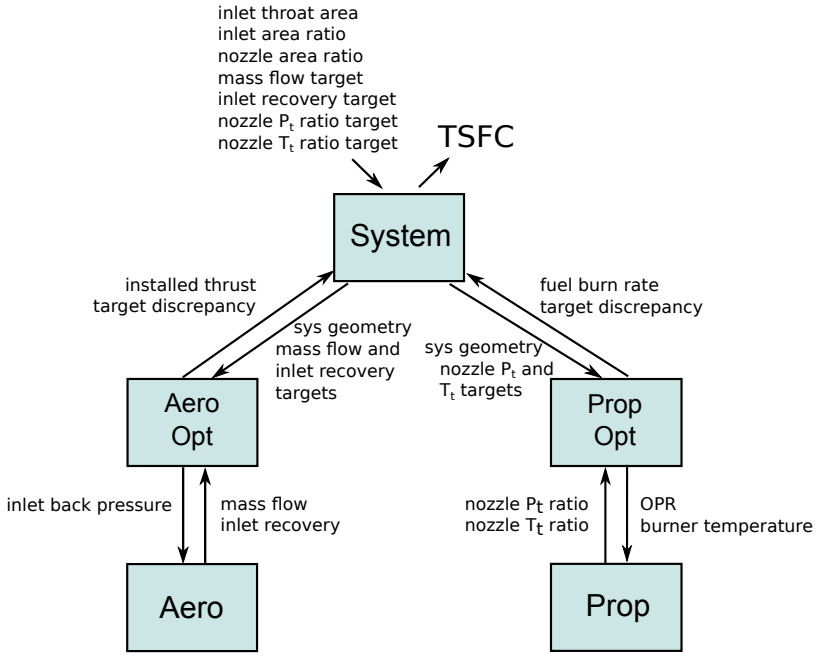

Figure 57: DSM for engine nacelle BCS

Note that there are many alternative ways to do the bi-level decomposition. For

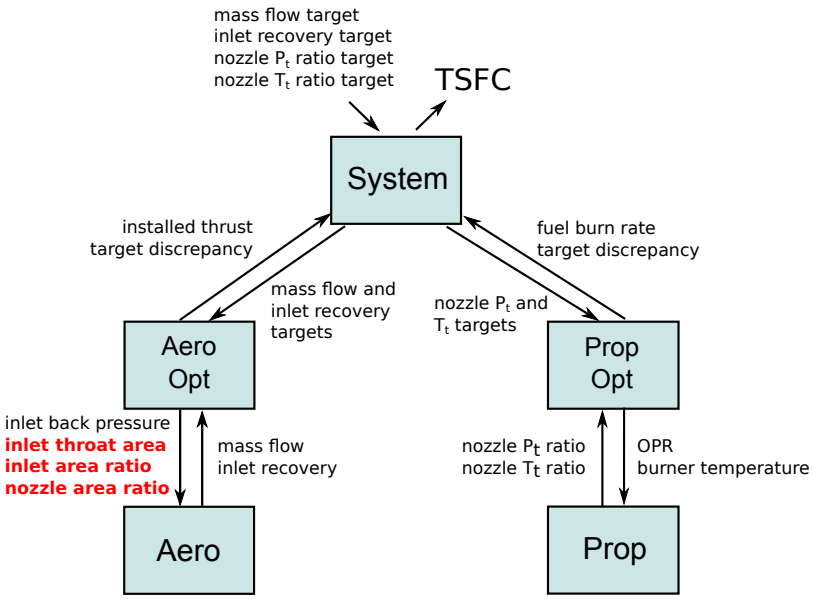

Figure 58: Alternate DSM for engine nacelle BCS

example, most of the nacelle geometry variables could be designed in the local aerodynamic subspace. This would result in a problem formulation like Figure 58. The reason for choosing the first bi-level decomposition has to do with the efficient reuse of CFD meshes and initial solutions. Mesh generation and connectivity comprise a significant fraction of the total CFD cost. Also, the initial start-up of the CFD inlet and nozzle boundary conditions is computationally expensive. Because of the very high flow gradients, the inlet and nozzle flows often require slow local time stepping and gradual enforcement of boundary conditions. For example, the nozzle plenum pressure may have to be started at free-stream values and gradually increased to nominal values. To establish basic flow features such as the exhaust plume, external shocks and inlet separation, the CFD may have to run for several flow-through times (time required to traverse the flow domain at freestream velocity). This may be alleviated using multi-grid and grid cycling techniques, but it is still a major cost for problems with strong inlet and outlet flows. The Python scripts controlling OVERFLOW runs contains routines to set time steps and CFL numbers and adjust them in response to

convergence problems that typically occured in the early iterations of a simulation.

If all of the geometry variables are at the system level, the local search at the aerodynamic discipline subspace only changes flow boundary conditions, such as inlet back pressure and nozzle plenum temperature. The discipline level search only requires a single grid based on system variables and a single initial solution. Once major CFD flow structures are established, the same grid and solution may be reused with modified flow conditions for several disciplinary iterations in a given system iteration. This is much more efficient than re-gridding and initializing a solution for every single CFD function call.

In a (very approximate) analogy, one may think of the cost of starting a physical turbine engine from free stream conditions; it may be difficult to coax the engine to life and establish running steady conditions. Once the engine is running smoothly, it may be relatively easy to change its operating conditions. Since CFD also has a major numerical starting cost but can have its flow boundary conditions changed relatively easily, there are more or less efficient ways to exploit this in the choice of bi-level decomposition.

The BCS bi-level decomposition of variables is given below.

$$\mathbf{Z}_{\text{system}} = \begin{bmatrix} \text{inlet throat area} \\ \text{inlet area ratio} \\ \text{nozzle area ratio} \\ \text{mass flow target} \\ \text{inlet total pressure recovery target} \\ \text{nozzle plenum total pressure target} \\ \text{nozzle plenum total temperature target} \end{bmatrix}$$

$$\mathbf{X}^{\text{aero}} = \begin{bmatrix} \text{inlet AIP static back pressure} \\ \text{nozzle plenum total pressure}_{\text{ local}} \\ \text{nozzle plenum total temperature}_{\text{ local}} \end{bmatrix}$$

$$\mathbf{X}^{\text{prop}} = \begin{bmatrix} \text{weight} \\ \text{burner temperature}_{\text{ local}} \\ \text{overall pressure ratio}_{\text{ local}} \end{bmatrix}$$

Recall that each discipline takes system variables and computes local, discipline-level values for certain system target variables. For example, in the aerodynamics discipline, inputs include copies of system targets $\mathbf{Z}^{\text{aero}}_{\text{system}}$: mass flow and inlet P_t recovery targets. The local outputs $\mathbf{Z}^{\text{aero}}_{\text{local}}$ are actual mass flow and inlet P_t recovery from CFD. Then, the bi-level design formulation becomes:

System Level

$$
\begin{aligned}
\text{Minimize:} \quad & F_{sys} = && \text{installed thrust-specific fuel consumption (TSFC)} \\
& && \text{with constraint penalties} \\
\text{by sampling min. of:} \quad & && -\log E\left[I\left(F(\mathbf{Z}_{\text{sys}})\right)\right] - \sum_{k}^{K} \log P\left(-\epsilon \leq J_k\left(\mathbf{Z}_{\text{sys}}\right) \leq \epsilon\right) \\
\text{with respect to:} \quad & && \mathbf{Z}_{\text{sys}} \\
\text{subject to:} \quad & h_1 = && \text{thrust} - 10,000 \text{ lbf} = 0 \\
& h_2 = && \text{inlet mass flow} - \text{nozzle mass flow} = 0 \\
& 0.40 \leq && \text{Mach}_{\text{AIP}} \leq 0.55 \\
\text{and updating:} \quad & && \text{Bayesian models } p(F|\mathbf{Z}_{\text{sys}}), p(J_k|\mathbf{Z}_{\text{sys}}) \\
& && \text{for } k = 1, 2, \ldots K \text{ coupling variables} \\
\text{given:} \quad & && \text{Mach}_{\infty} = 0.8, \text{ steady cruise} \\
& && \text{altitude} = 30,000 \text{ ft}
\end{aligned}
$$

Aerodynamics Discipline

$$
\begin{aligned}
\text{Minimize:} \quad & Q_{\text{aero}} = \sum_{k}^{K_{\text{local}}^{(i)}} J_k^2 \\
\text{with respect to:} \quad & \mathbf{X}_{\text{local}}^{\text{aero}}
\end{aligned}
$$

Propulsion Discipline

$$
\begin{aligned}
\text{Minimize:} \quad & Q_{\text{prop}} = \sum_{k}^{K_{\text{local}}^{(i)}} J_k^2 \\
\text{with respect to:} \quad & \mathbf{X}_{\text{local}}^{\text{prop}}
\end{aligned}
$$

After formulating the bi-level problem, the next step is to describe the setup of discipline analysis codes, as highlighted in Figure 59.

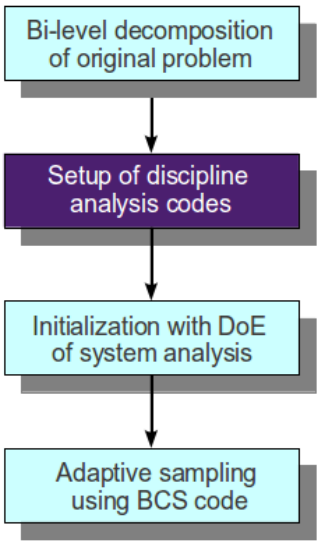

Figure 59: Bi-level decomposition is the first step in implementing BCS

6.1.3 Aerodynamics Analysis

CFD is done with the NASA OVERFLOW version 2.2b overset CFD code. Meshing is automated in Chimera Grid Tools using a Python script. Overset mesh connectivity information is computed using NASA's Pegasus 5.1. A Roe flux difference splitting upwind scheme and a Liou AUSM+ scheme were tried, but these proved to be sensitive to solver settings in the design space for this particular problem. In other words, settings that results in a stable solution at one design point may not produce a converged solution when used for another point in some other region of the design space. These settings include limiters for upwind Euler terms, dissipation, and smoothing parameters. Despite the higher potential accuracy of upwind schemes, a simple centered difference for Euler terms is used because of its robustness across the design space. This leads to more reliable convergence for different engine designs, provided that there are relatively high density meshes in plume/wake mixing regions and other high gradient regions. As before, a one equation Spalart-Allmaras

Table 14: OVERFLOW grid sensitivity for powered nacelle test case

mesh nodes (10^3)	η_p	error in η_p	thrust (lbf)	error in thrust
74	0.9409	-0.48%	9405.84	-0.97%
87	**0.9371**	**0.01**%	**9486.88**	**-0.12**%
153	0.9359	-0.05%	9499.50	0.01%
198	0.9357	-0.08%	9496.31	-0.02%
257	0.9360	-0.04%	9497.88	-0.005%
596	0.9364	—	9498.35	—

turbulence model was used. Figure 60 shows an example mesh.

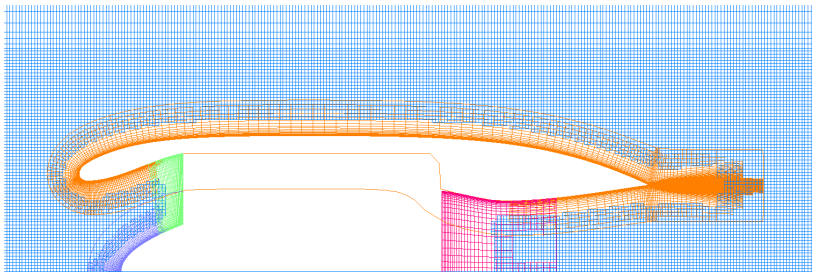

Figure 60: Chimera overset nacelle mesh (near-field view)

6.1.3.1 Grid Sensitivity

In the absence of public, experimental validation data, only a simple grid sensitivity study was done. The number of mesh nodes was varied primarily by increasing the boundary layer mesh density, the near field density, and the farfield distance. In all meshes, a maximum mesh expansion ratio of 1.2 was enforced. The most expensive case used a wall grid spacing of $y^+ = 1$ in the boundary layer, but this was increased in the coarser grids. Table 14 shows sensitivity information. The 87,000 node mesh was chosen for the BCS study. Figure 61 shows an example of such a mesh. The mesh is 13 body lengths in the axial direction (9 body lengths aft of the engine) and 3 lengths in the radial direction.

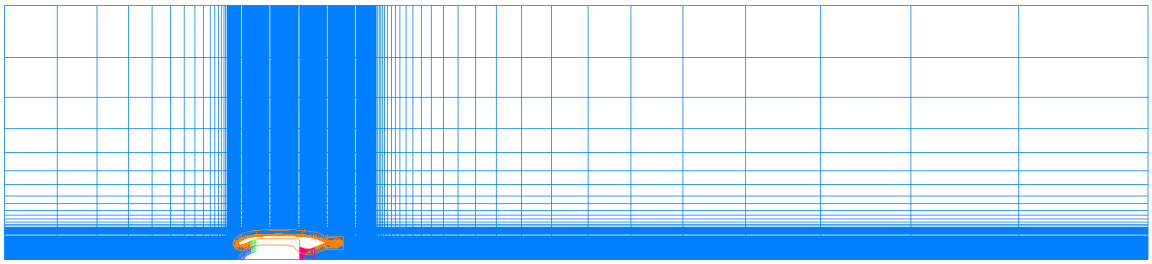

Figure 61: Engine nacelle mesh with 87,000 nodes

6.1.3.2 CFD Post-processing of System Level Outputs and Coupling Variables

Substantial effort was required to set up automated post-processing of CFD results. The installed thrust of the engine is computed by integrating the pressure, viscous stress, and momentum flux on a control volume enclosing the entire nacelle, AIP, and nozzle plenum and duct. This integration is again done with a dedicated Python script as well as the FOMOCO utility in OVERFLOW. Note that this approach does not distinguish between notions such as ram drag, uninstalled thrust, nacelle spillage drag, etc.

The main output coupling variables from the aerodynamics discipline are mass flow and inlet total pressure recovery. In CFD, two mass flows must actually be computed – for inlet and nozzle – and equality between the two must be enforced. The mass flow at the inlet is computed by integrating at the aerodynamic interface plane (AIP) in front of the compressor face. The inlet mass flow is controlled by a static back pressure condition at the AIP and calculated by a Python post-processing script. The nozzle mass flow is a more curious case. As mentioned earlier, given the throat area A* and plenum stagnation temperature and pressure, the nozzle mass flow reliably obeys a constant ratio of 0.735 with the one-dimensional isentropic value — to within three significant digits. Therefore, the nozzle mass flow is set by simply computing the isentropic critical throat area and enforcing this ratio.

Mass flow balance between the inlet and nozzle is enforced with a constraint

penalty added to the system objective function. This penalty is simply

$$(\text{nozzle mass flow} - \text{inlet mass flow})^2/\text{nozzle mass flow}$$

multiplied by a penalty weight (set to 10). As BCS progressively searches for lower system objectives, the mass flow imbalance is driven close to zero.

OVERFLOW and its force and moment integration utility (FOMOCO) do contain a convenient feature to enforce inlet-nozzle mass flow balance, but this feature is not used. In this feature, the inlet boundary conditions are automatically varied to make the inlet mass flow pursue the nozzle match flow. However, this feature cannot be easily used for BCS because the inlet-nozzle mass flow match may be physically infeasible for many of the sampled designs. For example, the inlet flow may choke so that its maximum value is below the nozzle mass flow. Then, the automatic inlet/outlet matching method in OVERFLOW/FOMOCO may not converge because the match is physically impossible. However, the adaptive sampling method requires *some* converged solution to advance the sequential learning process, even if it the inlet and nozzle mass flows do not match. For this reason, the inlet and nozzle mass boundaries are decoupled within the CFD analysis, but the mass flow balance is enforced outside of the CFD solver by a constraint penalty on the system objective.

The second major coupling variable is inlet total pressure recovery. The propulsion literature discusses several methods for expressing average P_t recovery over the AIP. Each of these have advantages and disadvantages, such as in the ways they conserve momentum or preserve entropy [123]. A mass average is chosen for its simplicity.

$$\text{Total pressure recovery} \quad \eta_p = \frac{\int\limits_{AIP} P_t \ (\rho u A) \ dA}{\int\limits_{AIP} (\rho u A) \ dA} \tag{74}$$

where P_t is total pressure, ρu is mass flux, and A is a dummy variable for area on the aerodynamic interface plane.

6.1.4 Propulsion Analysis

Engine cycle analysis is done with NASA's Numerical Propulsion System Simulation (NPSS). The code performs a "0-D" analysis that solves for properties, efficiencies, or conditions at stations in an engine's flowpath by enforcing mass, momentum, and energy conservation laws. The code is 0-D because it does not solve spatially discretized differential equations for the flow, but rather rather links flow components together and models the losses in each component by semi-empirical methods.

NPSS has a relatively flexible, object-oriented architecture that allows the solution of user-determined dependent variables with respect to independent variables. For this reason, the original DSM shown in Figure 53 is somewhat artificial. NPSS can be used to solve for output variables of nozzle total temperature and pressure given input variables of mass flow and inlet total pressure recovery. However, solver conditions can be changed so that nozzle stagnation conditions are enforced as inputs. This would lead to a one-way, feed-forward coupling, or an upper triangular DSM. Without a two-way coupling, the multidisciplinary analysis would not feature a fixed point iteration problem, removing the need for BCS altogether. Figure 62 shows the DSM with and without this two-way coupling.

However, the relatively simple design example is meant to represent more general, high-fidelity coupled aero-propulsion problems. If the propulsion cycle model uses high fidelity component models, then it would be much more difficult to fix nozzle conditions for and solve for dependents required for feed-forward multidisciplinary analysis. If the engine components were modeled using CFD, then this would require solution of an expensive inverse problem. For example, the inverse problem of finding a turbine geometry that results in a desired nozzle total temperature and pressure would likely involve an iterative solution with many CFD function calls.

The upper DSM in Figure 62 with the feedback is more representative of a typical, high fidelity aero-propulsion design problem. This structure is therefore used even

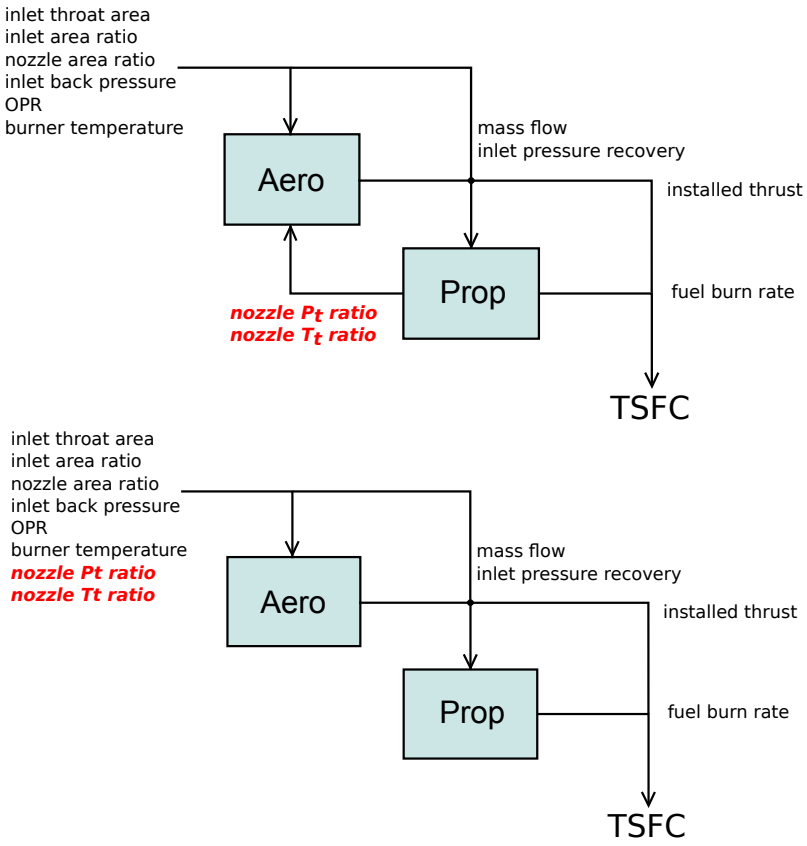

Figure 62: DSM for engine nacelle with and without feedback loop

though the complications of FPI could be averted by rearranging independent and dependent variables in NPSS.

6.1.4.1 Additional Coupling Mechanisms for Future Aero-Propulsion Examples

The powered nacelle example was chosen for its simplicity and similarity to the BLI aircraft example. Unfortunately for the goal of demonstrating BCS, the aero-propulsion coupling is weaker for the nacelle example. For example, the exhaust plume probably does not exert as strong an influence on the drag of nacelle body as it might on the airframe of the HWB. The axisymmetric nacelle inlet shape probably does not affect the total pressure recovery as strongly as the 3D HWB airframe and inlet geometry affect recovery and distortion.

In future work, additional important coupling effects could be included in a powered nacelle example if propulsion analysis includes higher fidelity models (either CFD

or empirical performance maps). First, the AIP Mach number likely has a strong influence on compressor efficiency. Currently, this influence is handled by assessing a penalty on the TSFC objective function for AIP/compressor face Mach numbers outside of a certain range. Second, the radial variation of properties on the AIP is likely an important coupling variable. Currently, the inlet total pressure recovery is expressed as a single, mass-flow average at the AIP. The inlet static back pressure is also enforced as a constant property over the AIP. Ignoring the radial variation may be acceptable for this simple demonstration, but it does not adequately capture an important mechanism for inlet total pressure loss. Figure 63 shows recirculating flow with stagnation pressure normalized by p_∞ near the inlet wall and AIP.

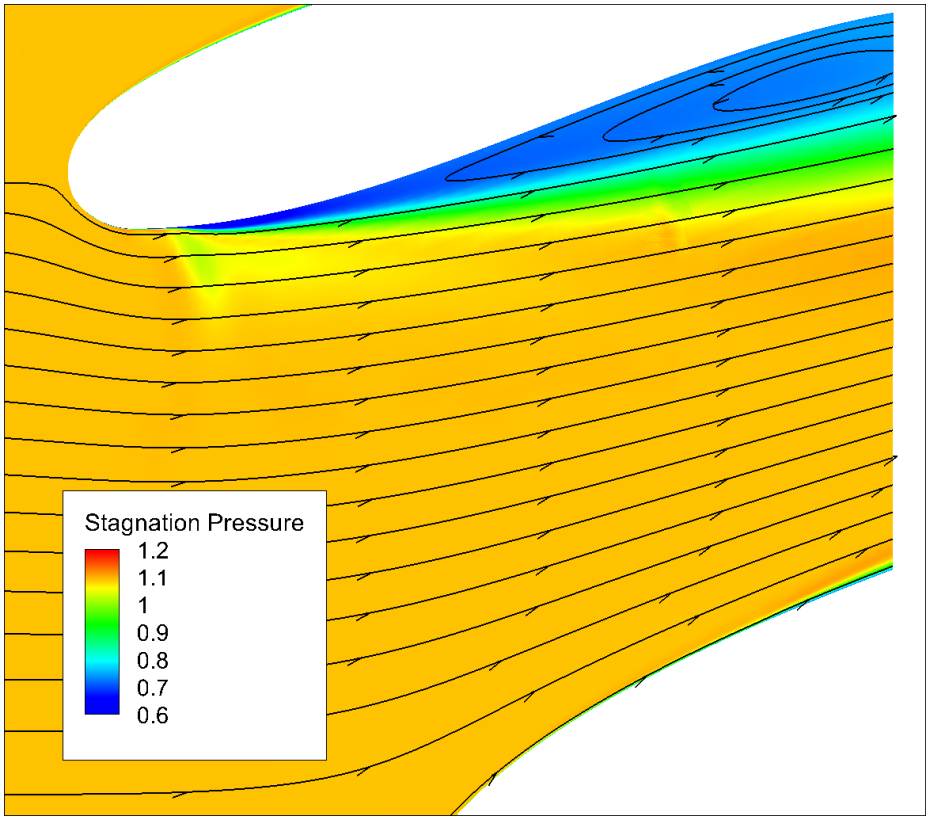

Figure 63: Recirculating flow and normalized stagnation pressure in inlet

As the air flow enters the inlet, the streamlines near the central axis encounter a pressure rise as they approach a stagnation point on the inlet cone/spinner. Most

of the compression of this air occurs externally due to this effect. From the stagnation region, this high pressure air encounters a relatively low pressure gradient as it approaches the compressor face. On the other hand, streamlines near the inlet lip encounter a sudden turn away from the original flow direction and then face a high adverse pressure gradient while approaching the compressor. This often leads to separation and a large loss of total pressure. In more realistic, high-fidelity design, the inlet back pressure would be parametrized to show radial variation. This reflects actual, physical designs, where the compressor blades would likely be designed to create radial pressure and mass flux variation to avoid the problems in Figure 63.

For these reasons, the designs sampled in this current study typically have relatively low inlet pressure recoveries, in the 90-93% range. However, the BCS method could still be used in future cases by adding radial variation to the coupling variables.

6.1.5 Warm-Start and Adaptive Sampling Details

After setting up the discipline analysis codes, the next steps are to initialize and run BCS until a computational budget is depleted.

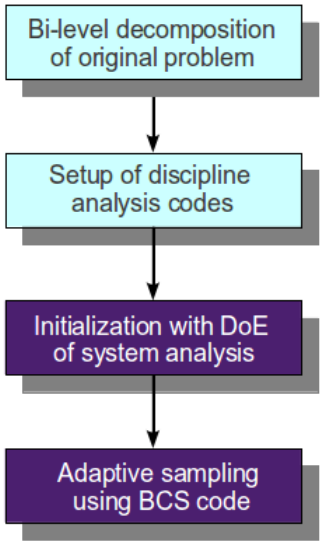

Figure 64: Warm-start and execution of BCS

While setting up the aerodynamics and propulsion codes and disciplinary optimizers, it was found that the propulsion code requires around three times more function calls to converge on minimum target discrepancies. However, the aerodynamics analysis is much more expensive, so it is again used as the main metric for computational cost.

The warm-start was done with a Latin hypercube DoE generated in Matlab. Because the number of CFD function calls per system call is not known a priori, a guessed value of 15 initial system DoE points was used because it was estimated to result in around 200 aerodynamics function calls. This proved to be a reasonable guess, as the DoE actually required 193 aerodynamics function calls and 743 propulsion function calls.

After this initialization, adaptive sampling with BCS was done until a budget of 1000 CFD function calls was depleted. This allowed 66 additional system sample points. Including the warm-start function calls, the final sample was 81 system sample points based on 1000 OVERFLOW CFD and 2944 NPSS function calls.

6.1.6 Engine Nacelle BCS Results and Discussion

Results from the BCS run are shown in this section. Recall that the intent of the last research question is to explore how BCS can be practically implemented with high fidelity, not to quantify its performance. In the absence of a very expensive direct optimization, there is no metric to show how accurately BCS performed. For example, the confidence measure $\delta_{95\%}$ is evaluated at the true optimum, which required 10^5 aerodynamic function calls in the earlier glider example.

CFD wall clock times were quite variable, but typically required several CPU minutes on Intel Westmere or i7 processors. On the other hand, a NPSS function call required less than a CPU second. Because the NPSS propulsion code is much faster than CFD, a relatively expensive but accurate settings for the optimizer (Matlab

fminsearch, simplex-based) were used for the propulsion disciplinary sub-problem, as opposed to a golden section line search for aerodynamics.

Figure 65 shows the BCS sample in terms of three normalized system-level design variables. This shows 40 system sample points (a snapshot taken when the BCS trial was half complete); circled points are adaptively sampled with BCS while 15 uncircled points are the warm-start points. The 40 system iterations required 511 CFD and 1676 NPSS function calls. Figure 66 shows the final sample of 81 system points that correspond to 1000 CFD and 2944 NPSS function calls.

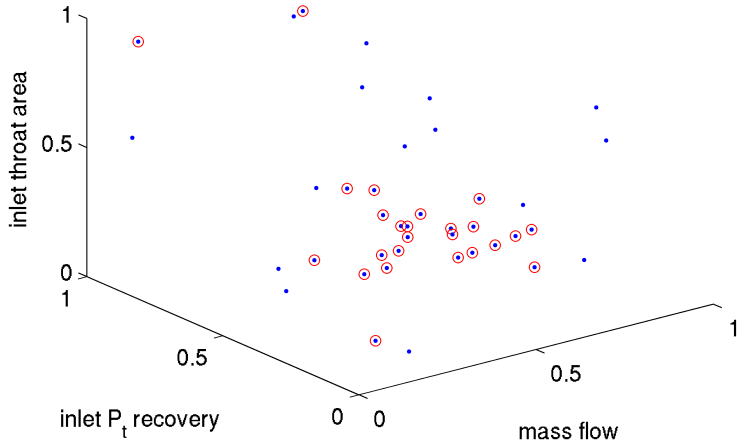

Figure 65: Sample of normalized design variables after 40 total system iterations (511 CFD, 1676 NPSS function calls); 15 blue dots are from warm-start, 35 red circled points are adaptively sampled

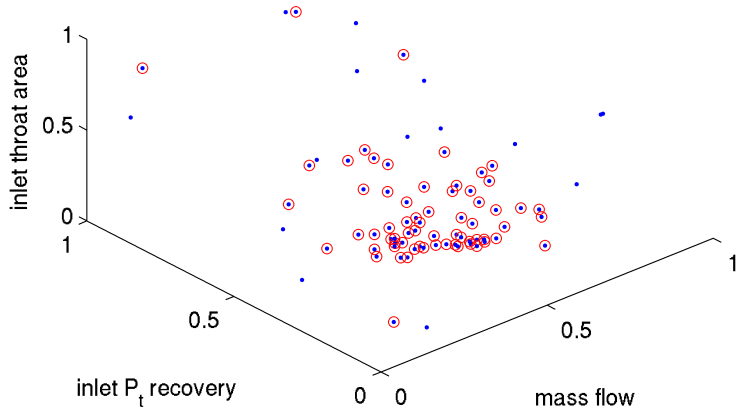

Figure 66: Sample of normalized design variables after after 81 total system iterations (1000 CFD, 2944 NPSS function calls); 15 blue warm-start points, 35 red circled adaptively sampled points

From Figure 66, it can be seen that sample points are concentrating in a region of the design space. Two dimensional Figures 67 to 69 show the sample in terms of one system design variable concentrating in regions of low objective values. Figures

70 through 73 show that most of the adaptively sampled points are concentrated in regions where target discrepancies J_k are small. These are but a few representative figures. Not all system targets and variables are shown because of the large number of combinations. Note that the points in these plots show adaptively sampled (circled) points clustering near $J_k = 0$, indicating low target discrepancy.

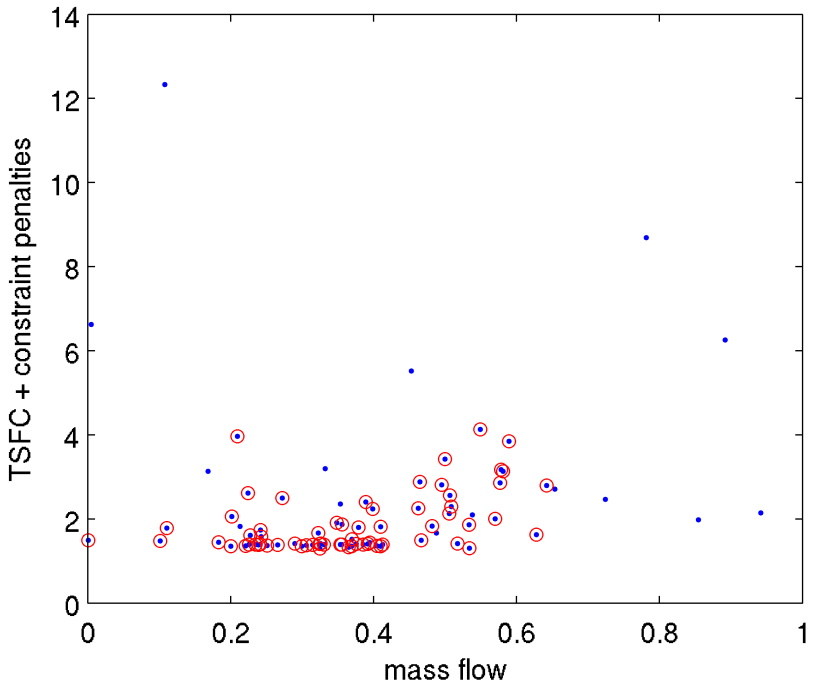

Figure 67: Objective vs. mass flow (normalized) for 81 system sample points (1000 CFD function calls)

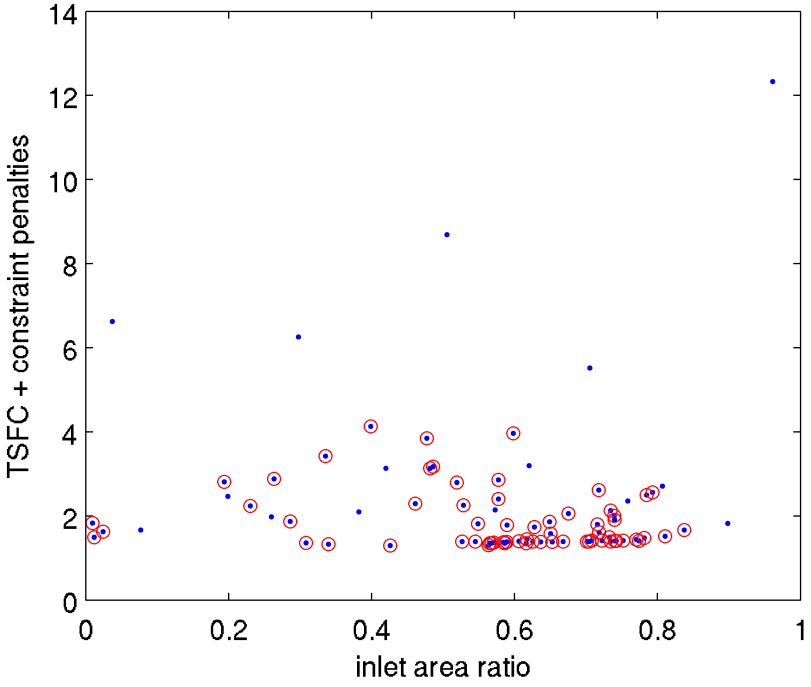

Figure 68: Objective vs. inlet throat area (normalized) for 81 system sample points (1000 CFD function calls)

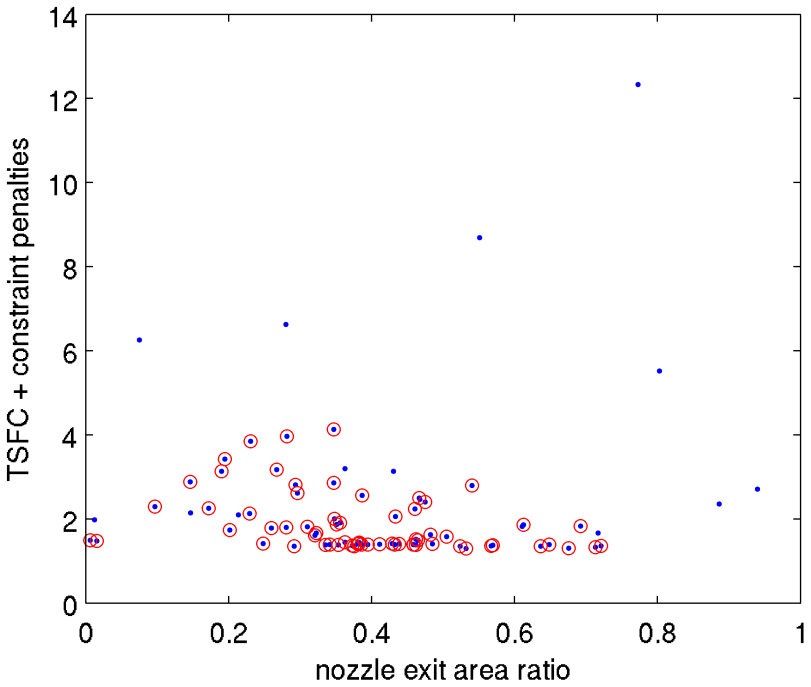

Figure 69: Objective vs. nozzle exit area (normalized) for 81 system sample points (1000 CFD function calls)

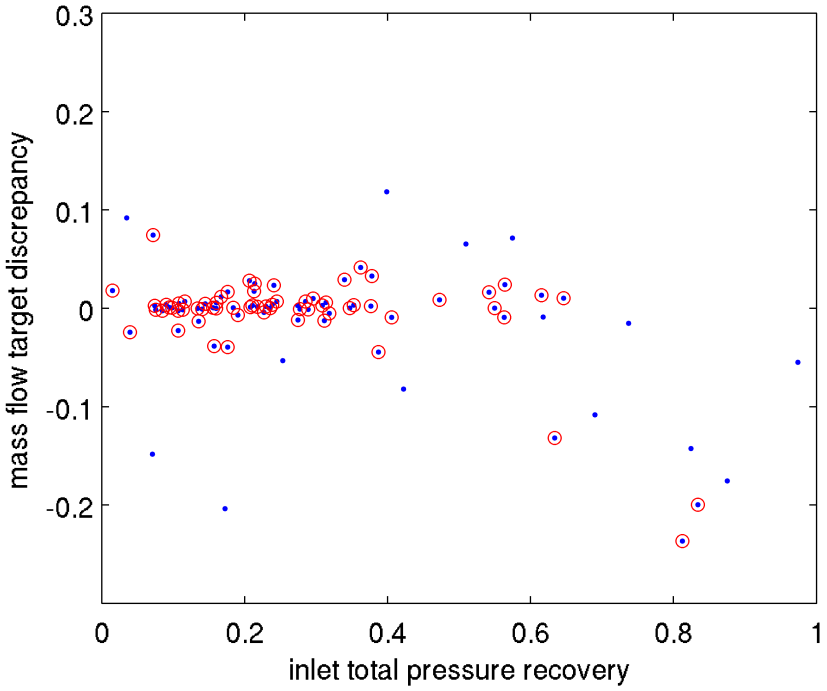

Figure 70: System-level mass flow target discrepancy vs. discipline-level inlet P_t recovery (normalized) for 81 system sample points (1000 CFD function calls)

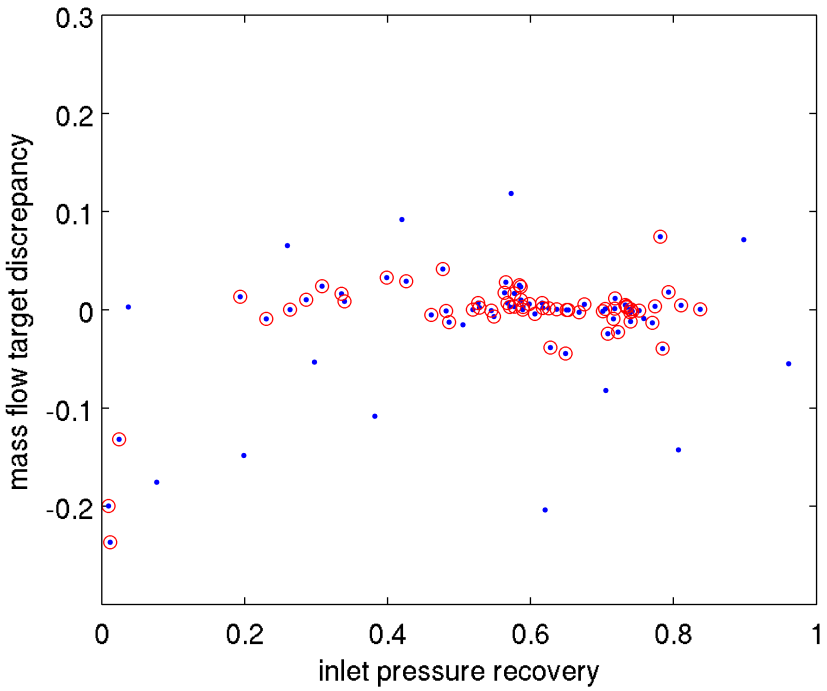

Figure 71: System-level mass flow target discrepancy vs. discipline-level inlet area ratio (normalized) for 81 system sample points (1000 CFD function calls)

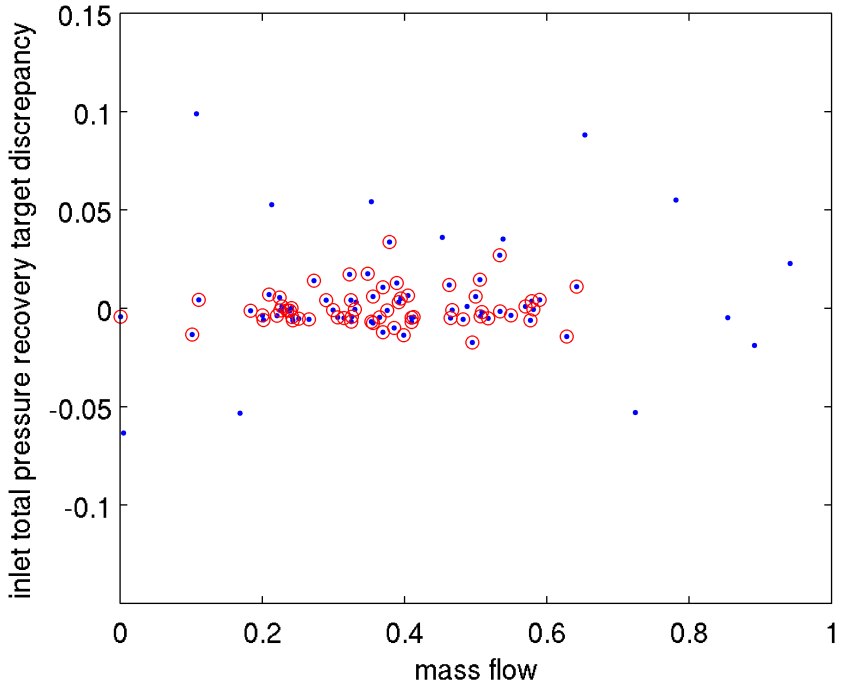

Figure 72: System-level inlet P_t recovery target discrepancy vs. discipline-level local mass flow (normalized) for 81 system sample points (1000 CFD function calls)

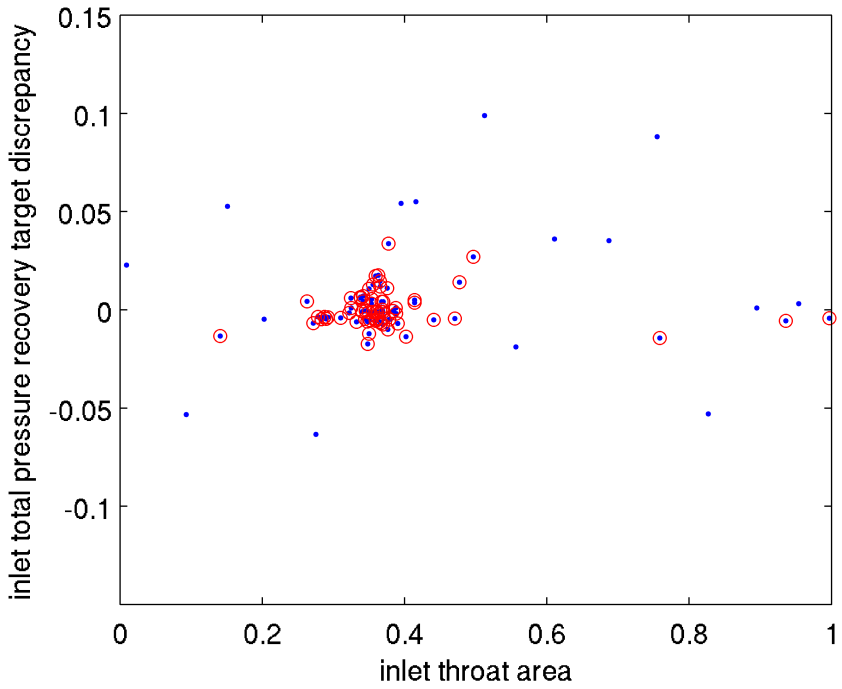

Figure 73: System-level inlet P_t recovery target discrepancy vs. discipline-level inlet throat area (normalized) for 81 system sample points (1000 CFD function calls)

Viewing the sample points gives the designer confidence that BCS is concentrating analysis resources according the intended criteria of 1) favorable objective and 2) low system target discrepancy. Note that a designer would have no guarantee that the concentrated regions are near the true compatible optimum. However, considerable insight about the design can be gained by Bayesian inference. The following figures show the compatible expected improvement criterion from the sparse Bayesian models of objective F and target discrepancies J_k:

$$-\log E\left[I\left(F(\mathbf{Z})\right)\right] - \sum_k \log P\left(-\epsilon \leq J_k\left(\mathbf{Z}\right) \leq \epsilon\right) \tag{75}$$

Plots are shown for two input variables in Figures 74 and 75. All other input variables that are not shown are set to the last sample point.

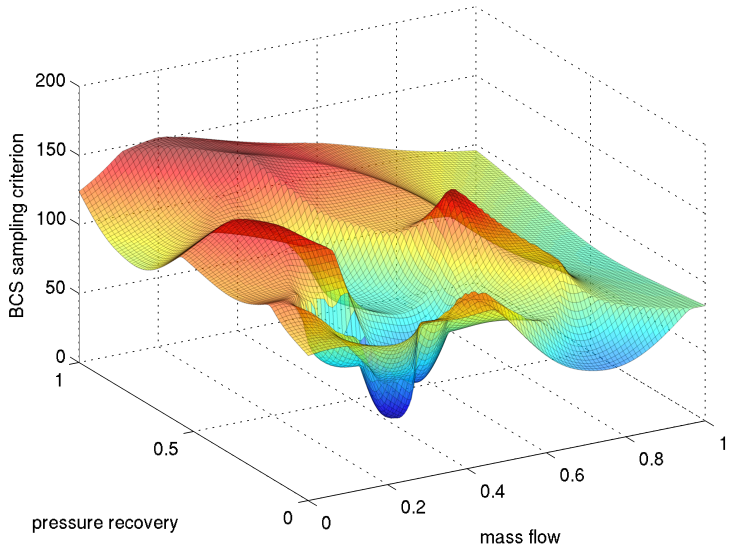

Figure 74: System-level sampling criterion (500 CFD function calls)

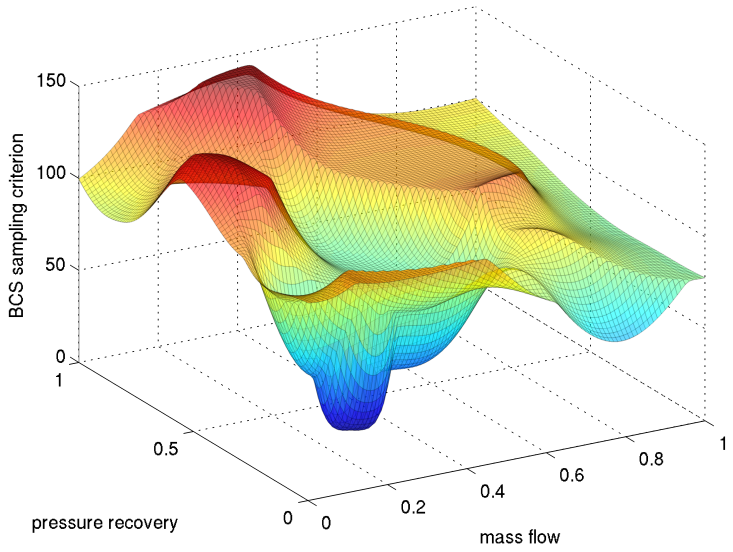

Figure 75: System-level sampling criterion (1000 CFD function calls)

It can be seen that the low, favorable region is relatively sharply defined by 500 CFD function calls. This region shifts slightly in the 1000 function call case in Figure 75 mainly because the other design variables (not shown) have slightly different settings. Suppose that with 1000 points, the engineer has depleted the computational budget for CFD. At this stage in the design process, there may be neither hope nor

need for finding the true optimum. The low, blue region in Figure 75 may be the best that can be done with 1000 points, balancing between local accuracy and global exploration. The designer can next study the favorable region and dissect the physical trends that create this low trough.

This dissection is easy because the BCS process requires separate Bayesian models for F and each target discrepancy J_k used in Equation 75. The engineer might view only the probability of compatibility, or the negative log probability that all target discrepancies lie within some tolerance, as in Eq. 76. In the case of Figure 76, the tolerance is set to $\epsilon = 10^{-3}$.

$$-\sum_k \log P\left(-\epsilon \leq J_k\left(\mathbf{Z}\right) \leq \epsilon\right) \tag{76}$$

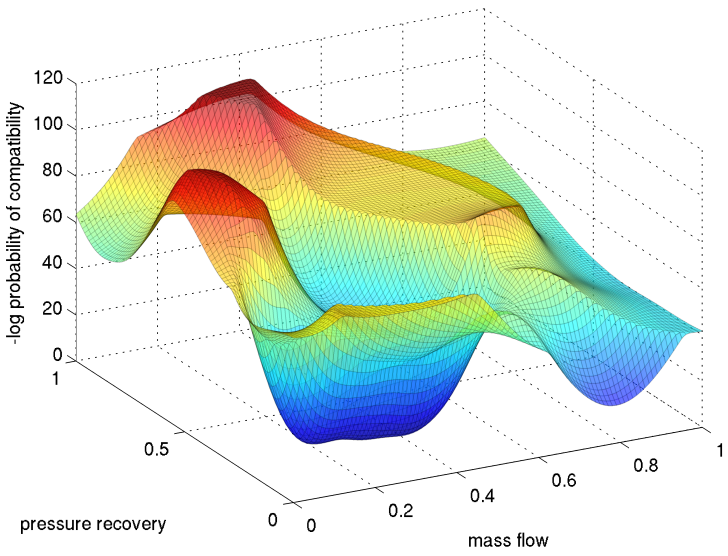

Figure 76: Probability of compatibility (1000 CFD function calls)

In addition, the designer may view an individual target discrepancy, J_k. For example, Figure 77 shows the predictive probability of mass flow discrepancy $J_{\text{mass flow}}$ with $\pm$ 1 standard deviation.

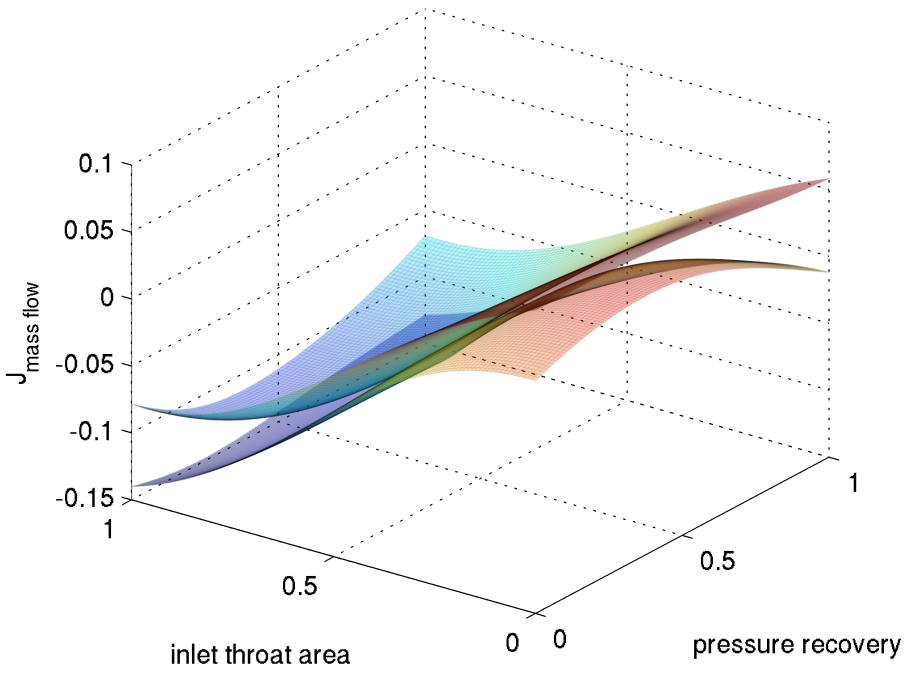

Figure 77: Predictive probability on mass flow discrepancy, $p(J_{\text{mass flow}})$, for $\pm$ 1 standard deviation (1000 CFD function calls)

After viewing these plots, the designer may want to learn about the physical causes of incompatibility or target discrepancy. For example, viewing sample points from regions of high mass flow discrepancy $J_{\text{mass flow}}$ reveals that designs with high system mass flow targets and small inlet throats tend to have choked inlet flow. Investigating regions of high discrepancy J_{recovery} show that high inlet area ratios and high inlet P_t recovery targets are incompatible, because these designs often have separated/recirculating flow and small shock waves in the inlet. Figure 63, the figure of recirculating flow that was shown earlier, is one such example.

Often, the close examination of the compatible regions and neighboring incompatible points will reveal unfavorable physical phenomena that are based on a bad choice of parameterization of the design problem. For example, the recirculating flow may point to an incompatible combination of mass flow, inlet throat, inlet area ratio, or AIP back pressures. However, it may be due to design parameterization or bad

default settings of detailed variables. This is similar to the problem encountered earlier in the BLI aircraft example, where nacelle shape parameters can influence shock formation that affects the entire aircraft. After investigating the results of BCS, the designer may select a particular design that resides in the favorable regions or adjust design ranges and add more detailed variables.

The BCS demonstration with a powered engine nacelle shows how an adaptive sampling for both a favorable objective and compatibility can help the designer narrow the design space and see the physical mechanisms for interdisciplinary compatibility. Much information about early design can be gleaned from a relatively meager budget of 500 to 1000 CFD function calls. The earlier glider test case suggests that a similar amount of information might not be gained from off-line DoE/surrogate modeling or direct optimization. This engine nacelle study showcases the practical implementation and advantages of the Bayesian approach.

Having described the results and insights from the engine nacelle test case, the discussion shifts to preliminary work on a boundary layer ingesting aircraft. This example is much more complex and computationally costly than the engine example, and the initial setup of the problem for BCS reveals some important final lessons for the designer.

6.2 Preliminary Findings: Boundary Layer Ingesting (BLI) Hybrid Wing-Body

Hybrid wing-body aircraft have wings that blend into a center body that also produces significant aerodynamic lift. In the recent years, design and research groups such as NASA, Boeing, MIT and Georgia Institute of Technology have studied the hybrid wing-body (or "blended wing body") concept as a replacement for current tube-and-wing configuration for subsonic cargo/transport aircraft [107], [74]. Potential advantages of this concept are greater cargo volume and higher lift/drag ratio. Boundary layer ingestion technology may further improve HWB aerodynamics to

increase fuel efficiency.

6.2.1 Boundary Layer Ingestion

BLI engines improve aircraft propulsive efficiency by re-energizing the wake. The following description is influenced by explanations by [12], [41], and [144].

Consider the idealized cases of (1) a podded engine well separated from the aerodynamic body, and (2) a boundary layer ingesting that is completely immersed in the wake of the body. In Figure 78, the grey regions represent wakes and the blue shaded regions represent velocity differences from freestream velocity at the Trefftz plane — the plane perpendicular to the wake and far downstream. In the first case, the wake of the aircraft leaves a velocity (or momentum) defect far downstream, while the engine plume leaves a velocity (or momentum) surplus. In the second case with the ideal wake-filling, BLI engine, this defect and surplus cancel out. This effect is described in greater detail.

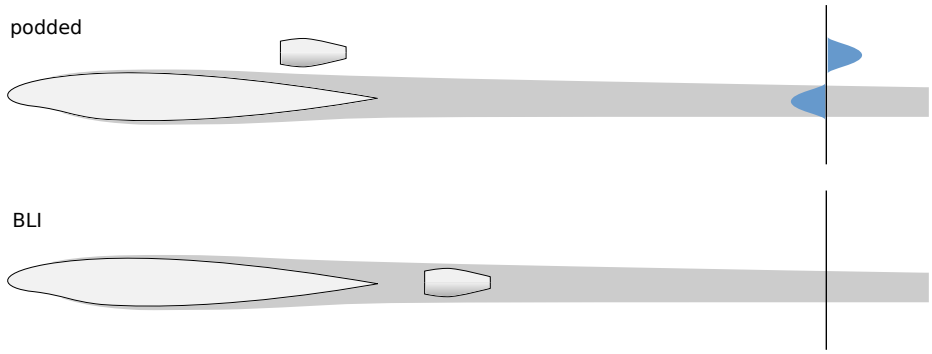

Figure 78: Podded and BLI aircraft and wakes

First consider the podded engine.

$$\begin{aligned} \text{Let } u_\infty &= \text{freestream velocity} \\ u_{\,\text{jet}} &= \text{average jet velocity} \\ u_{\,\text{wake}} &= \text{average aircraft wake velocity} \\ \dot{m} &= \text{mass flow rate} \end{aligned}$$

The thrust-drag balance of the entire podded configuration is given by

$$\begin{aligned} T_{\text{required}} &= D_{\text{aircraft}} \\ \dot{m}\,(u_{\,\text{jet}} - u_\infty) &= \dot{m}\,(u_\infty - u_{\,\text{wake}}) \end{aligned} \tag{77}$$

and the useful power for overcoming drag at freestream speed is

$$\begin{aligned} P_{\text{useful}} &= D_{\text{aircraft}}\, u_\infty \\ &= T_{\text{required}}\, u_\infty \\ &= \dot{m}\,(u_{\,\text{jet}} - u_\infty)\, u_\infty \end{aligned} \tag{78}$$

The rate of work done by the engine on the ingested streamtube equals the change in kinetic energy of the air as it accelerates from freestream to jet speed.

$$P_{\text{podded}} = \frac{1}{2}\dot{m}u_{\text{jet}}^2 - \frac{1}{2}\dot{m}u_\infty^2$$

Substituting the T_{required} from (77), we have

$$P_{\text{podded}} = \frac{1}{2}\, T_{\text{required}}\,(u_{\,\text{jet}} + u_\infty) \tag{79}$$

Note that this amounts to the engine thrust multiplied by the average of streamtube velocity fore and aft of the engine. The propulsive efficiency η_{prop} is the ratio of useful power to engine power:

$$\eta_{\text{prop, podded}} = \frac{P_{\text{useful}}}{P_{\text{podded}}} \tag{80}$$

Now consider the BLI case. For simplicity, assume that the ideal case where engine ingests the entire wake of the aircraft and accelerates it exactly to freestream velocity, such that $u_{\text{jet}} = u_\infty$. Unlike the podded case in Eq (79), since the ingested streamtube is accelerated from wake speed to freestream speed. So the BLI engine power is

$$P_{\text{BLI}} = \frac{1}{2}\dot{m}u_\infty^2 - \frac{1}{2}\dot{m}u_{\text{wake}}^2$$

However, if we assume the aircraft drag is the same as before, then T_{required} from Eq.(77) remains the same. Substituting, we get

$$P_{\text{BLI}} = \frac{1}{2}\, T_{\text{required}}\, (u_{\text{wake}} + u_\infty) \tag{81}$$

Now, compare (79) and (81):

$$P_{\text{podded}} = \frac{1}{2}\, T_{\text{required}} \left(\underbrace{u_{\text{jet}}}_{} + u_\infty \right) \quad \text{versus} \quad P_{\text{BLI}} = \frac{1}{2}\, T_{\text{required}} \left(\underbrace{u_{\text{wake}}}_{} + u_\infty \right)$$

Since $u_{\text{jet}} > u_{\text{wake}}$, we can see that $P_{\text{podded}} > P_{\text{BLI}}$. For the same aircraft drag, the podded configuration requires more power. This can also be seen in terms of $\eta_{\text{prop,podded}} < \eta_{\text{prop,BLI}}$. This can also be understood intuitively: thrust is produced by imparting momentum (or velocity) to air. For the same velocity increase, it is more energy-efficient to increase velocity over a lower average value because energy depends on velocity squared.

The foregoing discussion is highly idealized, and it assumes that drag of the aircraft is well-defined and distinct from engine losses. However, if airframe and engine are

tightly integrated, it is often unclear where the airframe starts and the engine begins. The thrust-drag distinction is less meaningful and losses from airframe versus engine are mainly a matter of accounting. For example, it is unclear if the main advantage of BLI should be described as 1) engine plume filling the wake to reduce drag and required thrust for a *given engine power*, or 2) the wake decelerating air entering the engine to decrease power requirement for *given drag and thrust*. Recent work by Drela introduces a unified method of accounting for aerodynamic and propulsion losses in terms of mechanical power dissipation, avoiding arbitary separation of thrust and drag [41].

The wake ingestion principle is already used in some cruise missiles and torpedos, and is almost universal for ship propulsion. It is relatively difficult to exploit this effect in aircraft, because the wake is spread out due to dispersed wing and fuselage, and it is difficult to capture this wake in the streamtube of an engine. In addition, it has been difficult to devise design methods that exploit this phenomenon in early design to have a significant impact on final aircraft performance.

The propulsive advantage of BLI is countered by costs in thermodynamic efficiency. The main mechanisms for this cost are total pressure recovery and distortion at the fan or compressor face. Total pressure recovery η_p is defined as the ratio of total pressure P_t at the fan face and the freestream total pressure.

$$\eta_p = \frac{P_{t,\text{ fan face}}}{P_{t,\infty}} \tag{82}$$

It is a common measure of engine inlet efficiency. It is similar to a ratio of kinetic energies at the fan face and freestream, but it is conventionally used for compressible flows because of its ease of measurement and computation. Inlet total pressure recovery has a significant effect on the propulsion cycle, thermodynamic efficiency, and ultimately the thrust-specific fuel consumption (TSFC).

Inlet distortion is another major cost of BLI on engine performance. Distortion

typically refers to the variation of total pressure (or other property) at some aerodynamic interface plane (AIP) immediately in front of the fan or compressor face. One definition from Seddon and Goldsmith [134] is:

$$\text{Distortion} = \frac{P_{t,\max} - P_{t,\min}}{P_{t,\text{mean}}}$$

where P_t is total pressure. Distortion can refer to dynamic or steady effects, which both affect the propulsion cycle. In the context of BLI aircraft design, the term often focuses on circumferential variation. Distortion directly affects the thermodynamic cycle of the engine but also causes losses through unsteady effects on fan blades as they travel through a full revolution. This tends to decrease fan efficiency and can raise structural requirements because the fan blades encounter oscillating aerodynamic loads in the distorted air. Distortion is one of the main design challenges for BLI aircraft or combat aircraft for which inlet, engine, and airframe must be tightly integrated in a limited volume [68], [157]. Several recent papers such as Plas et al [12] and Florea et al [51] have investigated the impact of distortion and mitigation methods for BLI applications.

6.2.2 Aero-Propulsion Design of a BLI Aircraft

The boundary layer ingestion (BLI) concept involves engines placed close to the aircraft body in order to ingest boundary layer air. By ingesting the sluggish air into the engine and exhausting energized air into the aircraft wake, the concept can potentially achieve higher propulsive efficiency than conventional designs. This may outweight thermodynamic penalties for the propulsion cycle and yield greater fuel efficiency.

However, BLI leads to complications in traditional methods for conceptual/preliminary design of aircraft. In typical, traditional sizing and synthesis methods, constant lift-drag polars are often assumed. A mild aero-propulsion coupling occurs because the engine fuel burn affects weight lapse and required lift. Using empirical models or data for engine performance (an 'engine deck'), fuel weights can be varied to iteratively

solve for weight, drag, and thrust at various phases of a mission. For sizing, energy balance methods can also be used to estimate performance constraints for various mission phases or maneuver conditions, as described by authors such as Mattingly [70]. In such methods, aircraft aerodynamics and propulsion are highly decoupled to simplify early analysis.

However, BLI inherently relies on strong aero-propulsion coupling to achieve a performance benefit. From the propulsion engineer's point of view, the engine ingests "dirty" air, which results in a less efficient propulsion cycle. But due to the aerodynamic benefits of wake-filling, lower power is required, so there is lower absolute fuel burn. On the other hand, from the aerodynamicist's point of view, the drag polars' shapes depend on the engine's performance (assuming that airframe lift-drag polars can even be distinguished from engine thrust effects). The disciplines are mutually dependent so that, even if they show disadvantages in isolation, the combined effect can be advantageous for the entire system.

To maximize the benefits of BLI, the close coupling should be exploited in early design, when most expensive design commitments are made. This may include design decisions such as the number of engines, wing area or aspect ratio. But to capture the coupling effects, there must be accurate modeling of high fidelity physics. After all, to exploit boundary layer ingestion, it is helpful to have a boundary layer in one's physics model.

For these reasons, the BLI HWB can make a very good demonstration of BCS in particular and also show generally how to use expensive codes efficiently in highly coupled early design. Some recent efforts include low fidelity conceptual design efforts at NASA [107] as well as higher fidelity work by researchers at MIT [74],[12], and Rodriguez at Stanford [123]. The last author used CFD and a propulsion cycle code with fixed point iteration (FPI) for direct optimization of BLI and podded engine HWBs. The study uses direct, high-fidelity modeling of the boundary layer ingestion

and wake filling physics and their interactions with the engine cycle. But despite its recent publication date, the research work was conducted in the early 1990s with CFD meshes of about 600,000 cells. These meshes are relatively coarse compared to what computational budgets allow in 2011, so this valuable research can be extended by further study. Thus, it would be useful to study the BLI HWB both as a platform for BCS as well as for investigating the aircraft concept itself.

6.2.3 Computational Cost of BLI HWB Aerodynamics Analysis

A baseline N2B geometry was provided by Boeing and NASA under the NASA GSRP fellowship. A CAD model was refined before section coordinates were extracted and smoothed using a Python script. The HWB was represented parametrically as an airfoil stack with section sweep angles, dihedral angles, local twists and section spans. Winglets were not included, since they are detailed features that are usually designed for particular operating regimes. The parametric HWB geometry generator was linked to another series of Python scripts that control generation of a complete set of overset CFD meshes, mesh connectivity, and a CFD run file. The entire environment was run on the NASA Advanced Supercomputing (NAS) cluster, Pleiades, at NASA Ames Research Center. Typical CFD simulations used from 16 to 128 Intel Xeon Harpertown, Nehalem, or Westmere processors for $O(10)$ hours of wall clock time.

The structured, overset meshes were generated with Chimera Grid Tools (CGT) and overset mesh connectivity was computed with Pegasus 5.1 from NASA Ames. An example of the overset mesh is shown in Figure 79. The OVERFLOW 2.2 solver was used with a Roe upwind scheme, a Beam-Warming block tridiagonal scheme, and a one equation Spalart-Allmaras turbulence model. These tools have previously been validated in a variety of examples, including the transonic aircraft of the NASA Common Research Model (CRM) [133].

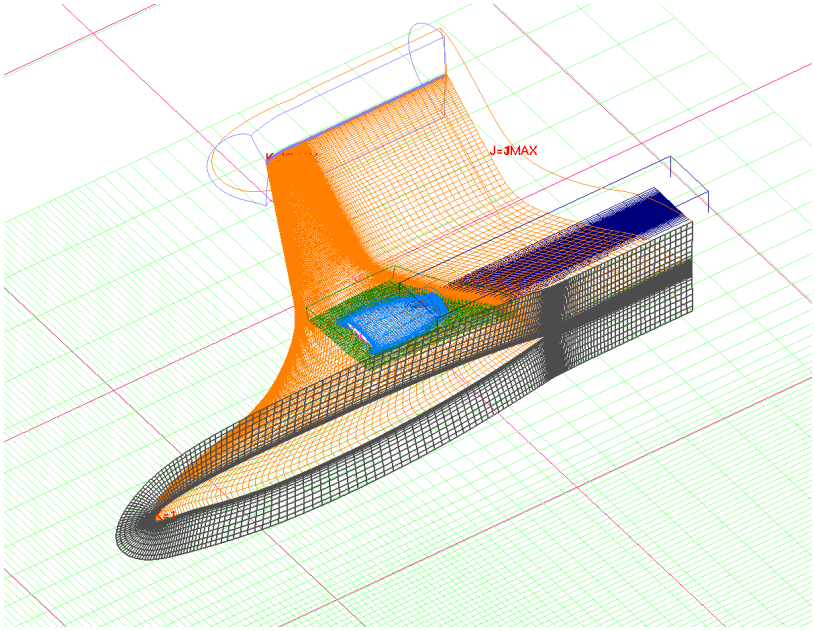

Figure 79: BLI HWB overset mesh

Table 15: OVERFLOW grid sensitivity for HWB clean configuration

mesh nodes (10^6)	C_L	C_D	change in C_D
1.5	0.107	0.00611	—
6.9	0.116	0.00727	+19%
10.7	0.116	0.00652	-10%
17.0	0.115	0.00601	-8%

A grid sensitivity study was done on only the clean configuration (no engine nacelle) to gauge the required computational cost. Results are shown in Table 15, where C_D error refers to the percent error compared to the most expensive mesh. In all cases, boundary layer mesh thicknesses were set by a flat plate $y^+ = 1$ estimate based on Reynolds number Re_x of the average chord, assuming the entire HWB is a wing. The simulations were run until C_D varied by less than 0.1% in the last 100 iterations. In all cases, residuals dropped by at least three orders of magnitude in the near-field grids (body, outer wing, wing tip). Load integration was performed using the FOMOCO utility in the OVERFLOW suite.

Note that in the last case with 17 million nodes, the solution was still very sensitive to mesh size. This case required 1088 CPU hours on 32 Intel Xeon E5472 Harpertown

3.0 GHz processors. In wall clock time, this cost 34 hours in addition to about 20 hours of queue time. Other tests with up to 34 million mesh nodes were attempted, requiring over 2000 hours without full convergence. Further tests were halted due to this extreme cost, as it was apparent that there were not enough hours to do a full BCS test. Parallel efficiency is expected to fall as more processors are used. Faster Westmere processors may have been significantly faster due to superior memory bandwidth, but the NASA supercomputing facility would have charged proportionally more from the computing budget (by a factor of 2.2). Based on the results, it is not unreasonable to guess that a high quality N2B solution with powered nacelle and exhaust plume will require over 1500 CPU hours. The wallclock time is even more sensitive to computational cost, because queue time for multi-user high performance computing resources usually depends on the size of the job.

The computational budget for this NASA-sponsored research was 200,000 CPU hours, so there was a budget for only a few hundred function calls, with no room to accomodate mistakes. It was decided that this budget was insufficient to give a practical demonstration of BCS or to learn clear results about the N2B design.

6.2.4 Preliminary Insights: High Fidelity and Geometric Detail

Although the HWB analysis was too computationally expensive, there are some insights from the example that are useful both for general Bayesian sampling of high fidelity analysis as well as for the particular BLI-HWB application. One of the main insights is the problem of arbitrary, detailed geometry features having a strong influence on early design.

First consider the example of a general aircraft design. For conceptual design, one may be concerned with very general variables such as wingloading and thrustloading. Or, if considering slightly more detail, one may also design wing area and aspect ratio. One may assume that the precise airfoil camber and thickness distribution are less

important than these conceptual or preliminary design variables. But assume that the physics of the problem are complicated, so that high fidelity analysis must be used. High fidelity analysis usually requires a commensurate *geometric* fidelity. For example, for energy-based sizing and synthesis, a point-mass "geometry" model may be used for the aircraft. If using potential flow, the wing may be treated as a Prandtl lifting line or vortex lattice on a camber line. But if using CFD, a complete surface geometry or outer mold line (OML) must be specified. There are many detailed variables like wing root fillets and tip surfaces that are assumed to be less important. But because of the detailed geometry requirement for high fidelity analysis, these detailed variables must be set to some default value rather than no value at all. If the design problem is well understood, it is relatively easy to set these defaults. For example, wing airfoils may be set to common transonic test sections such as the RAE 2822 or NACA six digit series airfoils. However, if the problem is a new design concept, the arbitrary settings may be crucial. If transonic wing design was a novel problem, setting an arbitrary airfoil to a flat plate would yield very bad performance, and this "unimportant" variable may overwhelm the influence of aspect ratio or wing area. When using high fidelity codes in early design, the designer faces a challenge to artfully minimize or mask the effect of these arbitrary, detailed settings.

In the glider example, the simplicity of the potential flow and beam theory analysis matched the lack of geometric detail. For example, there was no issue of how to set a default wing tip shape, because the wing was modeled as a thin vortex lattice along the mean camber line. But when CFD analysis was used in the HWB case, the discrepancy between high fidelity analysis and low fidelity geometry became apparent.

In the case of the BLI HWB, preliminary design may include variables such as:

- wing area and aspect ratio
- inlet and nozzle area and aspect ratio

- nacelle length and position
- mass flow
- fan pressure ratio

BCS or another coupled analysis method could be done with CFD and a propulsion cycle analysis code to size the airframe and engine. The inlet and nozzle areas and aspect ratios are included so that the main boundary layer ingestion phenomenon can be adequately captured in early sizing. The designer may wish to find out whether the nacelles must have low or high inlet aspect ratio (e.g. a flat "mail slot" inlet) in order to realize the theorized BLI benefits, and also to know whether a propulsion cycle can be designed for such a nacelle.

The initial plan was to set a default nacelle geometry. For example, the nacelle inlet and nozzle exit faces were ellipses. The nacelle lip shape was parametrized with a parabola. The outer surface of the nacelle was formed with splines that were blended using Laplacian smoothing. This design parameterization was intended to create a reasonable nacelle shape with a small number of important variables.

In actual execution, the impact of the default detailed shape may be quite significant. Figure 80 shows Mach number near the nacelle on a plane orthogonal to the span. (Jagged edges represent fringes of trimmed overset grids. The edges and different color transparencies are due to TecPlot software's representation of overset CFD solutions.) A strong shock is visible above the nacelle. The following pictures use medium grids of less than 6 million nodes because they are intended to show a possible effect rather than physically accurate solutions. The shock tends to occur as air that has accelerated around the nacelle lip decelerates back toward free stream conditions. As with transonic airfoils, the detailed shape of the leading edge and top surface play a strong role in controlling the amount of overspeed and shock strength.

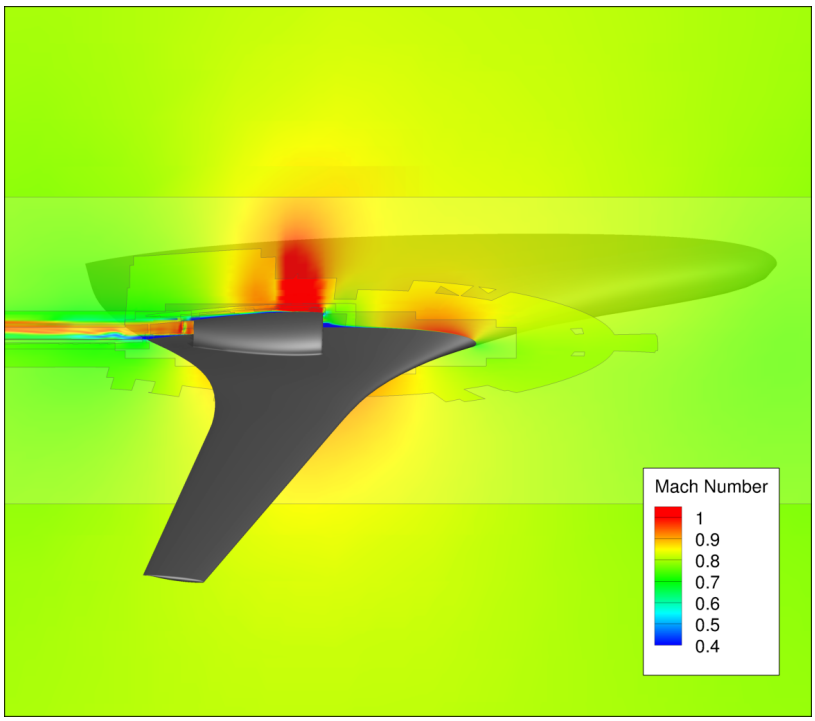

Figure 80: HWB; Mach number near nacelle

The shock is influenced by default detailed settings, but it affects the entire system's performance. Figure 81 shows normalized density near the aircraft surface. The effect of the shockwave is visible on outboard regions of the wing. Such shocks can cause a sudden adverse pressure gradient and induce boundary layer separation on the wing surface far from the nacelle. Figure 82 shows the same plot for a clean configuration without a nacelle, where the shock has different strength and structure on the wing surface. Such shocks tend to have a strong influence on drag because they can induce boundary layer separation. In this particular case, the nacelle shape has a significant influence on the aircraft performance, and this influence may be greater than simply the effects of boundary layer ingestion.

Once again, these solutions are not necessarily physically accurate (due to computational cost), and it is not known whether such shocks are typical in the design space. Yet the example notionally shows the problem that arises when high fidelity in early design requires high geometric detail that is not yet available for an immature

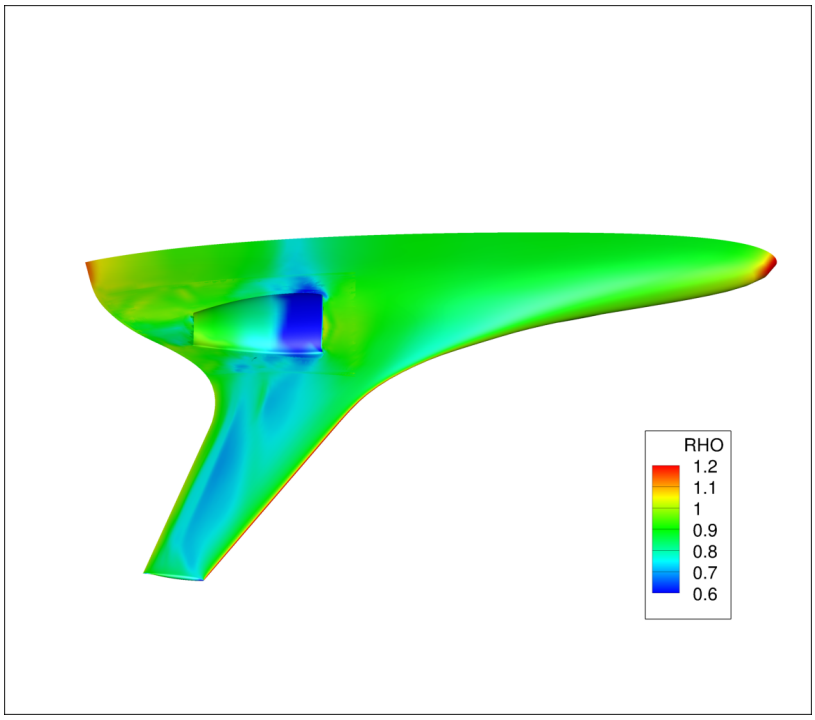

Figure 81: HWB with nacelle – density near surface

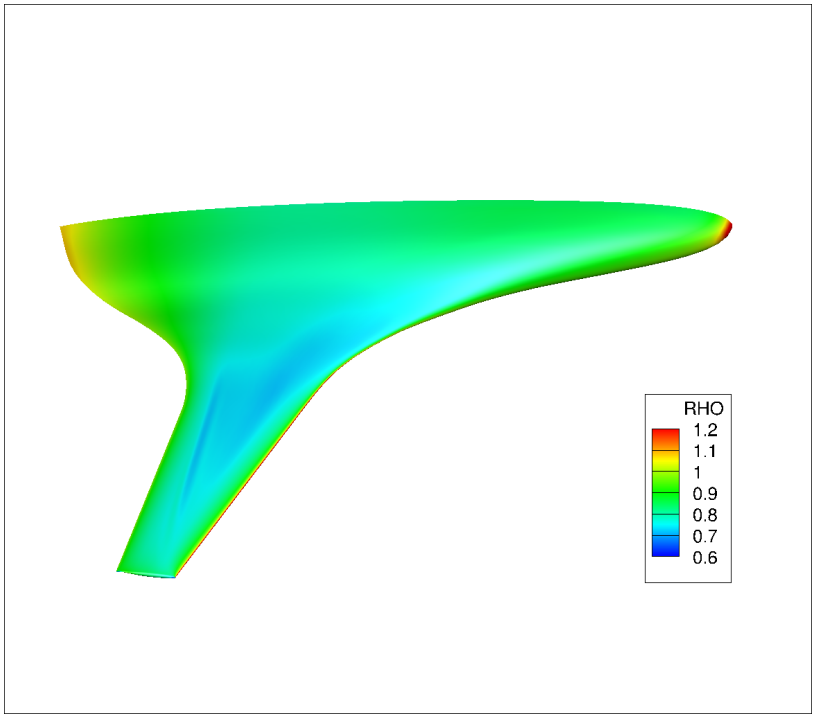

Figure 82: HWB clean configuration – density near surface

design. Because arbitrary settings for these details may come to dominate the design performance, part of the designer's art must be to creatively devise ways to minimize their influence. The mere availability of high fidelity features should not tempt the designer to use all of them.

For example, the main physics for early BLI HWB design may have been better captured by an actuator disc instead of a true nacelle geometry. Such a mesh patch could have boundary conditions that add total pressure and temperature to represent a nozzle and a static back pressure boundary condition to represent an inlet. Another option may be to add a nacelle shape with no lip or thickness and a frictionless slip wall to minimize the influence of the detailed nacelle shape. Still a third option might even be to deliberately choose local grid and solver settings that are highly diffusive to artificially smear out the influence of shocks. This may include the use of wall functions instead of dense boundary layer grids to minimize shock induced separation. Such options may or may not work. But the main point is that, if done carefully, a deliberately hobbled physics or geometry model may be more useful for early design than the highest available fidelity and geometric detail.

The BLI HWB is a promising concept that will likely continue to receive attention from the aerospace industry in the coming decades. It would be useful for future researchers to use BCS or a similar Bayesian sampling method to study early design. With rapid advances in computers, it is likely that a complete study with steady RANS CFD will be within the reach of academic researchers in a few years.

CHAPTER VII

CONCLUSIONS

7.1 General Remarks

After studying detailed, practical issues of BCS in a high fidelity example, it is necessary to take a broad perspective and recall the main point of this dissertation. This section steps through the four sets of research questions, hypotheses and predictions of this thesis; they are also summarized in an Appendix.

7.1.1 Research Question, Hypothesis, and Prediction 1

The basic motivation was to investigate a common challenge faced by designers: how can one use computationally expensive high fidelity analysis in early design? Common solutions such as off-line sampling and direct optimization are often too expensive, and strongly coupled disciplinary physics may magnify this expense. This first hypothesis is very general: such problems can be addressed by conducting adaptive sampling based on an objective function as well as interdisciplinary compatibility. To test the hypothesis, there is a general prediction: is it possible to show one working example of a design method that works according that principle?

Notice that the term "Bayesian collaborative sampling" is not yet mentioned, and there are no specific tests. But the subsequent tests on BCS show that it is indeed an example of the general strategy, and thereby satisfy the general prediction.

7.1.2 Research Question, Hypothesis, and Prediction 2

The second research question directly continues from Prediction 1. Does Bayesian collaborative sampling serve as a proof-of-concept method of the general strategy of adaptive sampling based on a system objective and interdisciplinary compatibility?

Note that this question is not concerned with whether BCS is the most efficient possible implementation of the strategy. Future work may build on BCS and improve performance and computational efficiency with alternate Bayesian sampling criteria or problem decomposition. However, this thesis uses BCS as the first example of a new type of multidisciplinary sampling methods.

Hypothesis 2 is essentially that BCS behaves according to the general principles claimed in Hypothesis 1: concentrating points in a compatible and favorable region by adaptive sampling. The detailed investigations show that BCS does work and that it works according to the principles claimed. Prediction 2 poses a specific test using Bayesian confidence metric $\delta_{95\%}$. The metric measures the predictive certainty of a Bayesian model fit to some sample, and it is evaluated at the optimum that is computed independently. Samples are produced using a reference Latin hypercube DoE and BCS. It is predicted that $\delta_{95\%}$ is lower for BCS than for the DoE. In other words, after separately using a direct optimizer to find the true, compatible optimum of a design problem, BCS leads to a more certain and accurate prediction at that reference point compared to an off-line DoE.

The results from a canonical glider wing test example confirm the prediction and thereby support Hypothesis 2. Because $\delta_{95\%}$ is lower for BCS than the DoE at the compatible optimum design, it is convincing that BCS works according to the general strategy of concentrating sample points according to both a system-level goal as well as interdisciplinary compatibility. The machinery of Bayesian inference makes this process possible by balancing between exploitation of high certainty regions and exploration of low certainty regions of the design space.

After addressing Hypothesis 2 and establishing that BCS works as claimed, the thesis turned to a third level of inquiry.

7.1.3 Research Question, Hypothesis, and Prediction 3

Research Question 3 asks: "What are major factors that affect BCS performance?" This question was asked instead of: "How well does BCS perform compared to other multidisciplinary design methods?" As discussed in Chapter 5, the direct comparison between BCS and other methods was avoided because such a contest would inherently be ill-posed.

This is an example of a general methodological problem that affects much of the contemporary research in the multidisciplinary analysis and optimization field. When comparing design methods, there are usually many internal parameters for each method, such as the choice of internal solvers or tunable parameters such as relaxation factors or the size of finite difference steps. Because of the large number of such parameters within each design method, the question of relative performance is usually underdetermined and has infinite possible answers. Stated differently, because of the number of arbitrary parameter choices required to implement each method, the amount of performance variation within each method may be comparable to that between methods. So, if BCS performs "better" than a single-level expected improvement method with direct fixed point iteration, it is difficult to establish how much of this difference is due to the essential formulation of each method rather than user-selected parameters. This methodological difficulty is magnified because it is often difficult for the researcher to avoid unintentionally tuning his/her proposed method to a engineering test problem. For example, even adjusting design variable ranges may accidentally favoring one method.

For these reasons, the thesis investigated more modest yet tractable questions about factors that affect BCS performance. The intention was to provide credible tests for limited questions that may aid a practitioner in deciding whether or not to use BCS for some application.

Hypothesis 3 therefore focuses on the effect of coupling bandwidth (number of

coupling variables) on the performance of BCS. This particular effect is investigated in detail because the bi-level decomposition in based on coupling variables becoming system variables is the hallmark feature of BCS and its parent method, collaborative optimization. It is hypothesized that as the number of coupling variables in the original design problem increase, these variables become system design variable and eventually incur the "curse of dimensionality."

The prediction that flows from Hypothesis 3 is that BCS performance decreases as the number of coupling variables increase. In addition to the $\delta_{95\%}$ metric from earlier, another metric, $\epsilon_{95\%}$, was introduced to explicitly measure the predictive uncertainty about interdisciplinary compatibility at the true optimum. For both metrics, lower values indicate better performance (lower uncertainty at the optimum). The previous glider test problem was repeated in slightly modified form. The coupling bandwidth was varied by the number of structural deflection parameters. This test was somewhat artificial, but the manner of varying coupling bandwidth had to be carefully chosen to have a minimum change on the essential physics of the problem. In other words, coupling variables cannot be haphazardly added in a way that changes the reference optimum. The results of these tests support the hypothesis that BCS performance degrades with coupling bandwidth significantly for a representative engineering problem.

In addition to the main coupling bandwidth test, additional tests were done to identify factors that affect BCS performance. The results show that the initial warm-start sample size has a strong influence. At one extreme, a very small warm-start results in BCS to perform badly because there are no initial trends on which to adapt. The adaptive sampling tends to behave as a sequential space-filling sampling algorithm or prefers vertices of the design space until enough patterns emerge to exploit effectively. At the other extreme, a very large warm-start sample tends to degenerate into an off-line DoE, because there is only a small remaining budget for

adaptive sampling. In the glider test problem, a good balance between these two trends was achieved at an intermediate warm-start size, but in general, the optimum warm-start size is likely to be highly problem dependent. The more important finding is that BCS dependence on the warm-start size shows the expected trade-off.

Finally, additional tests were done to investigate replication error. Because the initial warm-start sample is randomly generated and also because the particular BCS implementation uses an internal genetic algorithm, BCS yields different results for replicated trials. This is not surprising in cases like the glider problem, where $\mathcal{O}(100)$ initial points are randomly distributed over a large design space of over ten dimensions. The repeated trials show that there is significant replication error, but this is likely an unavoidable consequence of doing adaptive sampling with a small number of analysis function calls in a relatively large design space. This replication error would tend to be alleviated as the BCS sample size increases.

7.1.4 Research Question 4

The last research question was relatively open-ended: "Can BCS be implemented practically with high fidelity codes like CFD?" No specific hypotheses or predictions were tested; instead, a realistic example using CFD was used to demonstrate practical implementation of BCS and learn lessons of practical tradecraft for the designer.

For earlier research questions, the glider wing example was used as a low-cost platform for theoretical tests, because many of these tests required computationally expensive optimizations to provide reference values. Such theoretical tests with canonical examples are crucial, but MDO practitioners often encounter promising algorithms that thrive in the greenhouse of academic tests but wither in a practical engineering environment. A high fidelity CFD example was studied to address this concern. The turbojet engine nacelle example shows that the general strategy of BCS can be made to work with high fidelity codes with complications of grid generation,

solver stability, and numerical noise. Even without a reference optimum to compute metrics like $\delta_{95\%}$ and $\epsilon_{95\%}$, the engine example shows qualitatively that BCS can concentrate sample points based on a system objective and compatibility. The example also demonstrates how a practitioner can use the final sample and Bayesian models for early design. In particular, the Bayesian models can be decomposed according to individual compatibility metrics. For example, the designer can see predictive probabilities for designs meeting mass flow or total pressure recovery targets. By dissecting the sample points that correspond to low probability of compatibility or system target match, the designer can explore the physical reasons for incompatibility. This type of qualitative information is often more useful to the designer in early design rather than the actual optimum value.

In addition to the engine nacelle problem, a preliminary study of a boundary layer ingesting (BLI) hybrid wing-body (HWB) was done to prepare it for a BCS test. A CFD mesh sensitivity study showed that BCS on the BLI HWB example would likely exceed the computational resources for this thesis. Nonetheless, there are important lessons from preliminary work. The main insight involves the relationship between physical fidelity and geometric detail, which is a general concern when using high fidelity codes in early design. Higher fidelity analysis usually requires a commensurate level of geometric detail. In early stages, a design is immature and has many unknowns. The designer typically focuses on a few important variables like wing aspect ratio; "unimportant" variables such as airfoils must be set to reasonable default values. In a truly novel design problem, reasonable default values may not be known *a priori*. Yet, these variables must be set to some value, since the physics analysis may require a complete geometry, such as a water-tight surface grid. In cases like the nacelle of a transonic BLI HWB aircraft, seemingly minor variables may have an unexpectedly strong physical influence on the system performance. Therefore, it must be part of the designer's art to carefully arrange high-fidelity analysis in early

design to mask the effect of such minor variables. While setting up codes like CFD for methods like BCS, the designer may even need to deliberately hobble the physical model or sacrifice fidelity to keep arbitrary detailed variable settings from obscuring the main trends in the system objective and interdisciplinary compatibility.

7.2 Risks for the Practitioner

Having reviewed the research questions, hypotheses, and predictions, the formal tests of this thesis have been summarized. However, the following remarks may be helpful to the practioner who must weigh the risks of using BCS or a similar method for a real design problem. These apply not just to the particular BCS method, but to uncertainty-based adaptive sampling in general.

The engine nacelle design example is different from the earlier glider examples because a direct optimization was not performed. For the glider tests, the metrics $\delta_{95\%}$ and $\epsilon_{95\%}$ were evaluated at the optimum, which required expensive direct optimization. These metrics can give us some confidence that the adaptive sampling was indeed finding favorable, compatible regions of the design space. In the engine nacelle example, direct optimization is computationally out of reach. There is no assurance that the "promising" regions shown in the final Bayesian models actually correspond to the true optimum region. This is a more realistic situation that a designer would face.

The greatest risk of an adaptive sampling strategy is that the algorithm may cluster points in a favorable region that is far from the true optimum. The densely sampled region may have acceptable performance, but the real problem is that sample points are drawn away from the true optimum, giving lower certainty in the region than may be achieved with a simple DoE. The earlier glider tests may give some tentative assurance. But, it should also be noted that this risk is notionally similar to the risk of an optimizer converging on the wrong value or on a local minimum. Bayesian

adaptive sampling additionally has a mechanism for exploiting current trends and exploring uncertain regions, so it may have an intermediate level of risk between an off-line DoE and a pure optimizer.

A second type of risk has to do with the complexity of adaptive sampling methods leading to either human mistakes or unexpected behavior that is sensitive to human choices. For example, in the first two attempts at using BCS on the engine nacelle example, more than 2000 CFD function calls were wasted due to minor BCS setup mistakes and data parsing errors. Bayesian adaptive sampling is unavoidably more complicated than an off-line DoE, so there is more chance for human errors. Even with a mature adaptive sampling code, there would be many human-tuned parameters (such as the basis width or choice of kernels) that could lead to pathological behavior of the sampling algorithm. For example, a bad choice of basis functions may cause the sampling algorithm to strongly prefer vertices of the design space.

This risk due to complexity and human choices is significant. However, the risk due to human error can also be offset by human attention. In this thesis, the test cases were conducted in a "hands-off" manner, using the minimal amount of intervention for methodological credibility. The Bayesian models were not manually retuned even when sample points were obviously clustering in the wrong region (in the designer's opinion). However, the same research methodology requirements do not apply to the actual activity of design. Recall that the motivation behind adaptive sampling is to use expensive, high fidelity codes efficiently when accurate global models and precise optimization are too expensive. In practice, very expensive experiments or function calls tend to be monitored carefully by the experimenters. For example, the a full aircraft RANS CFD analysis can require $O(1000)$ CPU hours per function call. In the author's experience, it is somewhat implausible that a CFD engineer would begin a Bayesian sampling script and return several weeks later to view only the final results of several million CPU hours. Furthermore, Bayesian methods like BCS tend

to be transparent during the sampling process. Similar to the figures in the nacelle example, a designer can pause the sampling process midcourse and view the current Bayesian models and sample points. He/she can dissect predictive probabilities on either system outputs or individual target discrepancies. If this dissection reveals errors in the sampling process or model predictions that contradict physical intuition, the designer can manually adjust the Bayesian sampler or simply revert to a space-filling DoE. This flexibility may offset the risks of Bayesian sampling.

To summarize, even though the complexity of Bayesian adaptive sampling increases the risk of wasting computationally expensive function calls, the very expense of the function calls usually means heavy scrutiny and human-in-the-loop intervention for practical design. The skill and the expertise of the engineer thus plays a strong role in the true risk when using Bayesian adaptive sampling for early design.

7.3 Sparse Bayesian Models

In this thesis, much effort was devoted to describing Bayesian inference and sparse Bayesian models in particular. The BCS method does not formally rely on any specific Bayesian model type — in theory, it could work with Kriging or simple linear Bayesian models, for example. However, the speed of sparse Bayesian models and their mechanism of Ockham's razor is a key enabler for adaptive sampling in relatively large design problems. Kriging-based Bayesian adaptive methods have been well studied in engineering design literature, but computational expense is a major obstacle when extending MDO research to multi-disciplinary, multi-fidelity, or multi-objective problems. The sparse Bayesian models or relevance vector machine discussed in this thesis are not proposed as a comprehensive solution or "silver bullet." Like many statistical models, they can be highly sensitive to internal tuning parameters or choice of basis functions, and therefore rely on the engineer's experience for effective use. Kriging or other models may be a better choice for many applications. However, this thesis

investigated the sparse modeling method as a promising strategy for achieving speed while yielding reasonable overall performance of the BCS method.

7.4 Future Research Directions

While this dissertation makes contributions to Bayesian sampling of highly coupled multidisciplinary problems, its preliminary nature must be emphasized. BCS is by no means the most efficient implementation that is possible, nor is it globally applicable to all problems. Future research directions may lie along three major axes: research into different architectures, different sampling methods, and different applications.

First, recall that BCS was created by combining Bayesian adaptive sampling (pioneered by Kriging-based methods) and a particular MDO architecture, collaborative optimization (CO). Other architectures may perform better in different problems. Some candidate architectures may include concurrent subspace optimization (CSSO) [136], BLISS2000 [147], and analytic target cascading [100]. The designer's choice of architecture will likely depend on qualitative features rather than raw performance in test problems. For example, the BCS method is particularly appealing because of the autonomy of its disciplines. Like CO, the method may be preferable in scenarios where the existing structure of a design organization needs to be preserved, and the autonomy of skilled discipline-level experts is critical. Every practical design problem has unique features and challenges, so investigating Bayesian sampling methods with alternate architectures is a promising avenue of research.

Second, there is room for further research in Bayesian models (or even models that yield predictive distributions). A promising path may be to simply improve the speed of Kriging models, or create Kriging fitting algorithms that allow efficient parallel computing. Another path may be to extend the sparse Bayesian framework to model forms other than the linear $y = \mathbf{w}^T \boldsymbol{\phi}$ form used in the present research.

Third, BCS or similar methods can be used to study important engineering applications. Many recent, promising aerospace concepts rely on strongly coupled physics to achieve performance advantages. The boundary layer ingesting aircraft is one example, but there are others. For example, the design of unducted transonic rotors relies on highly interdependent aerodynamics, structures, and acoustics disciplines. Amongs these, aero-acoustics in particular may rely on high fidelity physics, which makes the Bayesian adaptive sampling approach appealing. In a more detailed design application, the serpentine duct of an embedded nacelle of a BLI aircraft can be coupled with the design of turbofan blades to minimize the thermodynamic losses and distortion. Other important highly coupled design problems include hypersonic air-breathing aero-propulsion design and the aero-thermal-structures design of cooled turbine blades. As computer technology and high fidelity simulation develop, designers will likely explore many new design concepts that exploit interactions between engineering disciplines.

APPENDIX A

BAYESIAN COLLABORATIVE SAMPLING (BCS)

The following is a summary of the BCS formulation, which decomposes a multidisciplinary design problem into a system and discipline level. $\mathbf{Z}$ is the system design variable that includes the design variables of the original design problem as well as compatibility targets for each discipline.

System Level Problem

Minimize:	$F(\mathbf{Z})$	(including constraint penalties)
by sampling min. of:		$-\log E\left[I\left(F(\mathbf{Z})\right)\right] - \sum_{k}^{K} \log P\left(-\epsilon < J_k(\mathbf{Z}) < \epsilon\right)$
with respect to:	$\mathbf{Z}$	(shared design variables and coupling variables)
subject to:	$Q_i(\mathbf{Z})$	$= 0$ for $i = 1, 2, \ldots, I$
		(target discrepancy of each discipline)
	$G(\mathbf{Z})$	≤ 0
	$H(\mathbf{Z})$	$= 0$
and updating:		Bayesian models $p(F\|\mathbf{Z}), p(J_k\|\mathbf{Z})$
		for $k = 1, 2, \ldots K$ coupling variables

Discipline Level Problem

Each discipline has produces local outputs $Z_{\text{k, local output}}$ that correspond to system targets $Z_{\text{k,sys}}$ for $k = 1, 2, ..., K_{\text{local}}^{(i)}$ locally relevant coupling variables. For each of

$i = 1, 2, \ldots, I$ disciplines:

Minimize:	$Q_i\left(\mathbf{x}_{\text{local}}^{(i)}\right)$	$= \sum_k^{K_{\text{local}}^{(i)}} J_k^2 \quad \text{for} \quad J_k = Z_{\text{k,sys}} - Z_{\text{k, local output}}$
with respect to:	$\mathbf{x}_{\text{local}}^{(i)}$	
returning:	Target discrepancies J_k	
	$F(\mathbf{Z}), G(\mathbf{Z}), H(\mathbf{Z})$, if calculated in the i-th discipline.	

APPENDIX B

RESEARCH QUESTIONS, HYPOTHESES, AND PREDICTIONS

The first research question, hypothesis, and prediction are general and do not lead to formal tests. They set out a general strategy for multidisciplinary design. Note that Bayesian collaborative sampling is not yet mentioned, since it is a particular implementation.

Research Question 1: (General)

How can high fidelity analyses be used for multidisciplinary design if they are too computationally expensive to create a globally accurate surrogate model or to perform direct optimization?

Hypothesis 1: (General)

A Bayesian adaptive sampling method can concentrate sample points in 'favorable' regions of the design space with feasible, low objective values and interdisciplinary compatibility. It can do this by sampling unfavorable regions only enough to establish that they are probably unfavorable.

Prediction 1: (General)

At least one such adaptive sampling method can be demonstrated in a practical multidisciplinary problem.

The second set examines Bayesian collaborative sampling as a particular example that satisfies Prediction 1.

Research Question 2:

Does the particular method, Bayesian collaborative sampling (BCS), serve as an example of the general sampling strategy described earlier?

Hypothesis 2:

The Bayesian collaborative sampling (BCS) method behaves as in Hypothesis 1 – it concentrates sample points in regions of favorable objective and interdisciplinary compatibility.

Prediction 2:

In a representative design problem, BCS shows higher certainty about a global, compatible optimum than a typical DoE in a test problem. Assume that Bayesian models of the system objective $t(\mathbf{x})$ are fit to DoE and BCS samples of equal size, so that each gives a predictive distribution $P(t(\mathbf{x}))$. A compatible global optimum $\hat{t}(\hat{\mathbf{x}})$ is also calculated by optimization with direct fixed point iteration. Let $\delta_{95\%}$ be a measure of Bayesian confidence defined by the probability

$$P\left[\hat{t} - \delta_{95\%} < t\left(\hat{\mathbf{x}}\right) < \hat{t} + \delta_{95\%}\right] = .95 \tag{83}$$

as shown in Fig. 83. Then $\delta_{95\%,\ \mathrm{BCS}} < \delta_{95\%,\ \mathrm{DoE}}$.

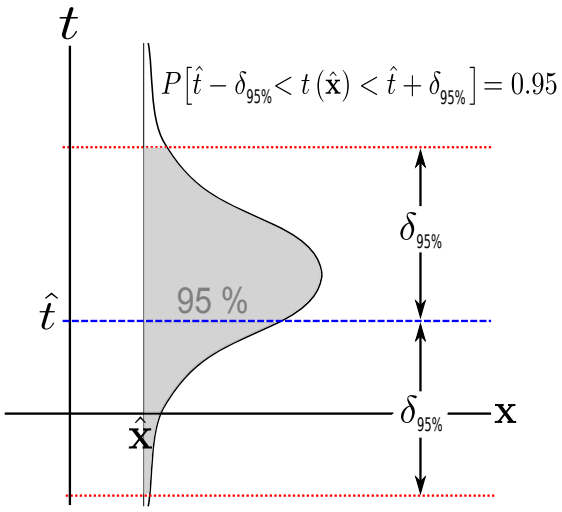

Figure 83: $\delta_{95\%,\ \mathrm{DoE}}$ as a measure of Bayesian confidence at the optimum

Research Question 3:

What are major factors that affect BCS performance?

Hypothesis 3:

Coupling bandwidth has a major effect on BCS performance. As the number of coupling variables increases, the system level of BCS increases in dimension and incurs the "curse of dimensionality."

Prediction 3:

BCS performance degrades as coupling bandwidth increases in one engineering test problem. Performance is measured by $\delta_{95\%,\,\mathrm{BCS}}$ and $\epsilon_{95\%,\,\mathrm{BCS}}$. $\epsilon_{95\%,\,\mathrm{BCS}}$ is measured by

$$P_{\text{compatibility}}(\epsilon_{95\%}) \approx \prod_{k}^{K} P\left(-\epsilon_{95\%} < J_k(\mathbf{Z}) < \epsilon_{95\%}\right) = 0.95 \tag{84}$$

for system design $\mathbf{Z}$ and target discrepancies J_k for $k = 1, 2, \ldots, K$ coupling variables.

Research Question 4:

Can BCS be implemented practically with high fidelity codes like CFD?

This last research question does not lead to formal hypothesis. It is deliberately open-ended and requires a practical implementation to learn lessons for the designer.

REFERENCES

[1] Agarwal, H. and Renaud, J. E., "New decoupled framework for reliability-based design optimization," *AIAA Journal*, vol. 44, pp. 1524–1531, July 2006.

[2] Aizerman, M., Braverman, E., and Rozonoer, L., "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and Remote Control*, vol. 25, pp. 821–837, 1964.

[3] Alexandrov, N., Dennis, J. E., Lewis, R. M., and Torczon, V., "A trust region framework for managing the use of approximate models in optimization," *Structural Optimization*, vol. 15, no. 2, pp. 16–23, 1998.

[4] Alexandrov, N. and Lewis, R., "Comparative properties of collaborative optimization and other approaches to MDO," in *Proceedings of First ASMO UK/ISSMO Conference on Engineering Design Optimization*, MCB Press, July 1999.

[5] Alexandrov, N., Lewis, R., Gumbert, C., Green, L., and Newman, P., "Optimization with variable-fidelity models applied to wing design," in *38th Aerospace Sciences Meeting and Exhibit*, (Reno, NV), AIAA, January 2000.

[6] Alexandrov, N., Lewis, R., Gumbert, C., Green, L., and Newman, P., "Approximation and model management in aerodynamic optimization with variable-fidelity models," *Journal of Aircraft*, vol. 38, pp. 1093–1101, 2001.

[7] Alexandrov, N., Nielsen, E., Lewis, R., and Anderson, W., "First-order model management with variable-fidelity physics applied to multi-element airfoil optimization," in *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, (Long Beach, CA), AIAA, September 2000.

[8] Alexandrov, N. M., *Multidisciplinary Design Optimization: State of the Art*, ch. Multilevel Methods for MDO, pp. 79–89. Society for Industrial and Applied Mathematics, 1997.

[9] Altus, S., Kroo, I., and Gage, P., "A genetic algorithm for scheduling and decomposition of multidisciplinary design problems," *Transactions of the ASME*, vol. 118, pp. 486–489, 1996.

[10] Anguita, D., Cravero, A., Filz, C., and Rivieccio, F., "An innovative technique for the aerodynamic design of turbine blade profiles using artificial intelligence," in *33rd AIAA Fluid Dynamics Conference and Exhibit*, (Orlando, Florida), AIAA, June 2003. SVM.

[11] Antelman, G., *Elementary Bayesian Statistics*. Edward Elgar Publishing, 1997.

[12] A.P. Plas, M.A. Sargeant, V. M. D. C. E. G. T. H. C. H., "Performance of a boundary layer ingesting (bli) propulsion system," in *45th AIAA Aerospace Sciences Meeting and Exhibit*, (Reno, Nevada), January 2007.

[13] Balabanov, V., Giunta, A., Golovidov, O., Grossman, B., Mason, W., and Watson, L., "Reasonable design space approach to response surface approximation," *Journal of Aircraft*, vol. 36, pp. 308–315, January - February 1999.

[14] Balling, *Multidisciplinary Design Optimization: State of the Art*, ch. Approaches to MDO Which Support Disciplinary Autonomy, pp. 90–97. Society for Industrial and Applied Mathematics, 1997.

[15] Balling, R. and Wilkinson, C., "Execution of multidisciplinary design optimization approaches on common test problems," *AIAA Journal*, vol. 35, pp. 178 – 186, January 1997.

[16] Barrett, T. R., Bressloff, N. W., and Keane, A., "Airfoil shape design and optimization using multifidelity analysis and embedded inverse design," *AIAA Journal*, vol. 44, pp. 2051 – 2060, September 2006.

[17] Batill, Stelmack, and Sellar, "Framework for multidisciplinary design based on response-surface approximations," *AIAA Journal of Aircraft*, vol. 36, Jan-Feb 1999.

[18] Batill, S. M., Renaud, J. E., and Gu, X., "Modeling and simulation uncertainty in multidisciplinary design optimization," tech. rep., AIAA, 2000.

[19] Bishop, C., *Pattern Recognition and Machine Learning*. Cambridge, U.K.: Springer, 2006.

[20] Bolstad, W. M., *Introduction to Bayesian Statistics*. J. Wiley, 2004.

[21] Branscome, C. and Mavris, D., "A regression confidence band approach to global optimization," in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Albany, New York), August-September 2004.

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

[23] Braun, R. D. and Kroo, I., *Multidisciplinary Design Optimization: State of the Art*, ch. Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment, pp. 98–116. Society for Industrial and Applied Mathematics, 1997.

[24] Calvetti, D. and Somersalo, E., *Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing*. Springer, 2007.

[25] Castonguay, P. and Nadarajah, S., "Effect of shape parameterization on aerodynamic shape optimization," in *45th AIAA Aerospace Sciences Meeting and Exhibit*, (Reno, Nevada), January 2007.

[26] Chen, S., Xiong, Y., and Chen, W., "Multiresponse and multistage metamodeling approach for design optimization," *AIAA Journal*, vol. 47, pp. 206–218, January 2009.

[27] Chiba, K. and Obayashi, S., "Data mining for multidisciplinary design space of regional-jet wing," *Journal of Aerospace Computing, Information, and Communication*, vol. 4, pp. 1019–1036, November 2007.

[28] Chiba, K., Oyama, A., Obayashi, S., Nakahashi, K., and Morino, H., "Multidisciplinary design optimization and data mining for transonic regional-jet wing," *Journal of Aircraft*, vol. 44, pp. 1100–1112, 2007.

[29] Choi, S., Alonso, J., Kroo, I., and Wintzer, M., "Multifidelity design optimization of low-boom supersonic jets," *Journal of Aircraft*, vol. 45, pp. 106–118, January-February 2008.

[30] Chung, J. and Lee, K., "Technical note: Shape optimization of transonic compressor blade using quasi-three-dimensional flow physics," *AIAA Journal*, vol. 40, pp. 389–391, February 2002.

[31] Clarke, S., Griebsch, J., and Simpson, T., "Analysis of support vector regression for approximation of complex engineering analyses," *Journal of Mechanical Design*, vol. 127, pp. 1077–1087, November 2005.

[32] Cosyn, P. and Vierendeels, J., "Numerical investigation of low-aspect-ratio wings at low reynolds numbers," *Journal of Aircraft*, vol. 43, pp. 713–722, May-June 2006.

[33] Cox, D. D. and John, S., *Multidisciplinary Design Optimization: State of the Art*, ch. SDO: A Statistical Method for Global Optimization, pp. 315–329. Society for Industrial and Applied Mathematics, 1997.

[34] Cox, R., "Probability, frequency, and reasonable expectation," *American Journal of Physics*, vol. 17, pp. 1–13, 1946.

[35] Cristianini, N. and Shawe-Taylor, J., *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[36] De Baets, P., Mavris, D., and Sobieszczanski-Sobieski, J., "Aeroelastic design by combining conventional practice with bi-level integrated system synthesis (bliss)," in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Aug 30 - Sep 1 2004.

[37] DeLaurentis, D., Lim, C., and Kang, T., "Approach to characterize mobility through modeling of solution spaces for conceptual aircraft," *Journal of Aircraft*, vol. 41, pp. 73–84, January-February 2004.

[38] DeLaurentis, D. and Mavris, D., "Uncertainty modeling and management in multidisciplinary analysis and synthesis," in *38th Aerospace Sciences Meeting & Exhibit*, (Reno, Nevada), AIAA, January 2000.

[39] Dennis, J. and Torczon, V., *Multidisciplinary Design Optimization: State of the Art*, ch. Managing Approximation Models in Optimization, pp. 330–347. Society for Industrial and Applied Mathematics, 1997.

[40] Drela, M., "Xfoil: An analysis and design system for low reynolds number airfoils," in *Low Reynolds Number Aerodynamics* (Brebbia, C. and Orszag, S., eds.), no. 54 in Lecture Notes in Engineering, pp. 1–12, Springer-Verlag, June 1989.

[41] Drela, M., "Power balance in aerodynamic flows," *AIAA Journal*, vol. 47, pp. 1761–1771, 2009.

[42] Drela, M. and Youngren, H., "Athena Vortex Lattice." http://web.mit.edu/drela/Public/web/avl/.

[43] Du, L., Choi, K. K., and Youn, B. D., "Inverse possibility analysis method for possibility-based design optimization," *AIAA Journal*, vol. 44, pp. 2682–2690, November 2006.

[44] Du, X. and Chen, W., "Concurrent subsystem uncertainty analysis and multidisciplinary design," tech. rep., AIAA, 2000.

[45] Du, X. and Chen, W., "Methodology for managing the effect of uncertainty in simulation-based design," *AIAA Journal*, vol. 38, pp. 1471–1478, August 2000.

[46] Du, X. and Chen, W., "Efficient uncertainty analysis methods for multidisciplinary robust design," *AIAA Journal*, vol. 40, pp. 545–552, March 2002.

[47] Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern Classification*. Wiley-Interscience, 2 ed., 2000.

[48] Fan, H.-Y., Dulikravich, G., and Han, Z.-X., "Aerodynamic data modeling using support vector machines," in *42nd AIAA Aerospace Sciences Meeting and Exhibit*, (Reno, Nevada), January 2004.

[49] Faul, A. C. and Tipping, M., "Analysis of sparse bayesian learning," in *Advances in Neural Information Processing Systems 14*, pp. 383–389, MIT Press, 2001.

[50] Felder, J., Kim, H. D., and Brown, G. V., "Turboelectric distributed propulsion engine cycle analysis for hybrid-wing-body aircraft," in *47th AIAA Aerospace Sciences Meeting*, (Orlando, Florida), January 2009.

[51] Florea, R. V., Reba, R., Vanslooten, P. R., Sharma, O., Stucky, M., O'Brien, W., and Arend, D., "Preliminary design for embedded engine systems," in *47th AIAA Aerospace Sciences Meeting*, (Orlando, Florida), January 2009.

[52] Forrester, Keane, and Sobester, *Engineering Design via Surrogate Modelling: A Practical Guide*. J. Wiley, 2008.

[53] Forrester, A., *Efficient Global Aerodynamic Optimisation Using Expensive Computational Fluid Dynamics Simulations*. PhD thesis, University of Southampton, 2004.

[54] Forrester, A., Keane, A. J., and Bressloff, N. W., "Design and analysis of noisy computer experiments," *AIAA Journal*, vol. 44, pp. 2331–2339, October 2006.

[55] Fukunaga, K., *Introduction to Statistical Pattern Recognition*. San Diego, CA: Academic Press, 2 ed., 1990.

[56] Gano, S., Renaud, J., Martin, J., and Simpson, T., "Update strategies for kriging models used in variable fidelity optimization," *Structural Multidisciplinary Optimization*, vol. 32, pp. 287–298, 2006.

[57] Gano, S. E., Renaud, J. E., and Sanders, B., "Hybrid variable fidelity optimization by using a kriging-based scaling function," *AIAA Journal*, vol. 43, pp. 2422–2430, November 2005.

[58] Giunta, A., *Aircraft Multidisciplinary Design Optimization Using Design of Experiments Theory and Response Surface Modeling Methods*. PhD thesis, Virginia Polytechnic Institute and State University, 1997.

[59] Glaz, B., Friedmann, P. P., and Liu, L., "Efficient global optimization of helicopter rotor blades for vibration reduction in forward flight," in *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Portsmouth, Virginia), September 2006.

[60] Gu, X., *Uncertainty in Simulation Based Multidisciplinary Design Optimization*. PhD thesis, University of Notre Dame, 2003.

[61] Gu, X. and Renaud, J., "Implementation study of implicit uncertainty propagation in decomposition-based optimization," tech. rep., AIAA.

[62] Gu, X. and Renaud, J., "Implicit uncertainty propagation for robust collaborative optimization," in *Proceedings of DETC '01, ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, (Pittsburgh, Pennsylvania), ASME, September 2001.

[63] Gunawan, S. and Papalambros, P., "A bayesian approach to reliability-based optimization with incomplete information," *Journal of Mechanical Design*, vol. 128, pp. 909–918, July 2006.

[64] Gunn, S. R., "Support vector machines for classification and regression," tech. rep., University with Southampton, May 1998.

[65] Hacioglu, A., "Fast evolutionary algorithm for airfoil design via neural network," *AIAA Journal*, vol. 45, pp. 2196–2203, September 2007.

[66] Huang, C., Galuski, J., and Bloebaum, C., "Multi-objective pareto concurrent subspace optimization for multidisciplinary design," *AIAA Journal*, vol. 45, pp. 1894–1906, August 2007.

[67] Huang, D., Allen, T., Notz, W., and Miller, R. A., "Sequential kriging optimization using multiple-fidelity evaluations," *Structural Multidisciplinary Optimization*, vol. 32, pp. 369–382, 2006. See citations for new sources on kriging and multifidelity optimization.

[68] Huenecke, K., *Modern Combat Aircraft Design*. Naval Institute Press, 1987.

[69] Hutchison, M., Unger, E., Mason, W., Grossman, B., and Haftka, R., "Variable-complexity aerodynamic optimization of a high-speed civil transport wing," *Journal of Aircraft*, vol. 31, pp. 110–116, January-February 1994.

[70] Jack D. Mattingly, William H. Heiser, D. T. P., *Aircraft Engine Design*. AIAA, 2002.

[71] Jameson, A. and Vassberg, J., "Computational fluid dynamics for aerodynamic design: Its current and future impact," in *39th AIAA Aerospace Sciences Meeting & Exhibit*, (Reno, NV), January 2001.

[72] Jaynes, E. T., "Bayesian methods: General background," *Introductory Tutorial, St. John's College and Cavendish Laboratory, Cambridge*, pp. 1–18, 1996.

[73] Jeong, S., Murayama, M., and Yamamoto, K., "Efficient optimization design method using kriging model," *Journal of Aircraft*, vol. 42, pp. 413–420, 2005.

[74] J.I. Hileman, Z.S. Spakovsky, M. D. M. S. A. J., "Airframe design for silent fuel-efficient aircraft," *Journal of Aircraft*, vol. 47, pp. 956–968, May-June 2010.

[75] Jin, Y., Olhofer, M., and Sendhoff, B., "On evolutionary optimization with approximate fitness functions,"

[76] Jones, D., "A taxonomy of global optimization methods based on response surfaces," *Journal of Global Optimization*, vol. 21, pp. 345–383, 2001.

[77] Jones, D., Schonlau, M., and Welch, W., "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, pp. 455–492, 1998.

[78] Jun, S., Jeon, Y.-H., Rho, J., and ho Lee, D., "Application of collaborative optimization using response surface methodology to an aircraft wing design," in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Albany, New York), August 2004.

[79] Keane, A. J., "Wing optimization using design of experiment, response surface, and data fusion methods," *Journal of Aircraft*, vol. 40, pp. 741–750, July - August 2003.

[80] Keane, A. J., "Statistical improvement criteria for use in multiobjective design optimization," *AIAA Journal*, vol. 44, no. 4, pp. 879–891, 2006.

[81] Keane, A. J. and Nair, P. B., *Computational Approaches for Aerospace Design: The Pursuit of Excellence*. New York: Wiley, 2009.

[82] Keane, A. J. and Petruzzelli, N., "Aircraft wing design using ga-based multi-level strategies," in *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, (Long Beach, California), AIAA, September 2000.

[83] Kennedy, M. C. and OHagan, A., "Predicting the output from a complex computer code when fast approximations are available," *Biometrika*, vol. 87, no. 1, pp. 1–13, 2000.

[84] Kim, H., Ragon, S., Soremekun, G., Malone, B., and Sobieszczanski-Sobieski, J., "Flexible approximation model approach for bi-level integrated system synthesis," Tech. Rep. 2004-4545, AIAA, 2004.

[85] Knight, M. and Wenzinger, C. J., "Wind tunnel tests on a series of wing models through a large angle of attack range: Part i – force tests," Tech. Rep. 317, National Advisory Committee for Aeronautics (NACA), 1928.

[86] Kodiyalam and Sobieszczanski-Sobieski, "Bilevel integrated system synthesis with response surfaces," *AIAA Journal*, vol. 38, pp. 1479–1485, Aug 2000.

[87] Kroo, I., *Multidisciplinary Design Optimization: State of the Art*, ch. MDO for Large-Scale Design, pp. 22–44. Society for Industrial and Applied Mathematics, 1997.

[88] Kroo, I., "Multidisciplinary optimization applications in preliminary design – status and directions," tech. rep., AIAA, 1997.

[89] Kroo, I. and Manning, V., "Collaborative optimization: Status and directions," in *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, (Long Beach, California), AIAA, September 2000.

[90] Leary, S., Bhaskar, A., and Keane, A. J., "Global approximation and optimization using adjoint computational fluid dynamics codes," *AIAA Journal*, vol. 42, pp. 631–641, March 2004. Last page has comments about computational cost of Kriging.

[91] Lian, Y. and Liou, M.-S., "Aerostructural optimization of a transonic compressor rotor," *AIAA Journal of Propulsion and Power*, vol. 22, pp. 880–888, 2006.

[92] Lin, J. G., "Analysis and enhancement of collaborative optimization for multidisciplinary design," *AIAA Journal*, vol. 42, pp. 348–360, February 2004.

[93] MacKay, D. J., "Bayesian interpolation," tech. rep., California Institute of Technology, Pasadena, California, May 1991.

[94] MacKay, D. J., "Information based objective functions for active data selection," *Neural Computation*, vol. 4, pp. 589–603, 1991.

[95] MacKay, D. J., *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[96] Madsen, J. I., Shyy, W., and Haftka, R. T., "Response surface techniques for diffuser shape optimization," *AIAA Journal*, vol. 38, pp. 1512–1518, September 2000.

[97] Martin, J. D. and Simpson, T. W., "Use of adaptive metamodeling for design optimization," in *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, (Atlanta, Georgia), September 2002.

[98] Martins, J., Alonso, J., and Reuther, J., "High-fidelity aerostructural design optimization of a supersonic business jet," *Journal of Aircraft*, vol. 41, pp. 523–530, May-June 2004.

[99] Mavris, D., Bandte, O., and DeLaurentis, D., "Robust design simulation: A probabilistic approach to multidisciplinary design," *Journal of Aircraft*, vol. 36, no. 1, pp. 298–397, 1999.

[100] Michelena, N., Park, H., and Papalambros, P., "Convergence properties of analytical target cascading," *AIAA Journal*, vol. 41, no. 5, p. 897905, 2003.

[101] Migdalas, A., Pardalos, P. M., and Varbrand, P., eds., *Multilevel Optimization: Algorithms and Applications*. Kluwer Academic Publishers, 1998.

[102] Mockus, J., *Bayesian Approach to Global Optimization*. Kluwer Academic Publishers, 1989.

[103] Montgomery, Myers, and Anderson-Cook, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. New York: Wiley, 2009.

[104] Nair, P. B., Choudhury, A., and Keane, A. J., "Bayesian surrogate modeling of deterministic simulation codes for probabilistic analysis," Tech. Rep. 2001-1676, AIAA, 2001.

[105] Nair, P. B. and Keane, A. J., "Coevolutionary architecture for distributed optimization of complex coupled systems," *AIAA Journal*, vol. 40, pp. 1434–1443, July 2002.

[106] Nair, P. B. and Keane, A. J., "Stochastic reduced basis methods," *AIAA Journal*, vol. 40, pp. 1653–1664, 2002.

[107] Nickol, C. L. and McCullers, L. A., "Hybrid wing body configuration system studies," in *47th AIAA Aerospace Sciences Meeting*, (Orlando, Florida), January 2009.

[108] O'Hagan, A., Kennedy. M. C. and Oakley, J.E., "Uncertainty analysis and other inference tools for complex computer codes (with discussion)," *Bayesian Statistics 6*, pp. 503–524, 1999.

[109] Ong, Y. S., Nair, P. B., and Keane, A. J., "Evolutionary optimization of computationally expensive problems via surrogate modeling," *AIAA Journal*, vol. 41, pp. 687–696, April 2003.

[110] Oyama, A. and Liou, M.-S., "Multiobjective optimization of rocket engine pumps using evolutionary algorithm," Tech. Rep. TM-2001-211082, NASA, 2001.

[111] Papila, M. and Haftka, R. T., "Uncertainty and response surface approximations," Tech. Rep. 2001-1680, AIAA, 2001.

[112] Peigin, S. and Epstein, B., "Computational fluid dynamics driven optimization of blended wing body aircraft," *AIAA Journal*, vol. 44, pp. 2736–2745, November 2006.

[113] Perez, Renaud, and Watson, "Adaptive experimental design for construction of response surface approximations," Tech. Rep. 2001-1622, AIAA, 2001.

[114] Perez, V. M., Renaud, J. E., and Gano, S. E., "Constructing variable fidelity response surface approximations in the usable feasible region," Tech. Rep. 2000-4888, AIAA, 2000.

[115] Pritesh C. Mody, Sho Sato, D. K. H. e. d. l. R. B. J. I. H. E. W., "Conceptual design of an n+3 hybrid wing body subsonic transport," in *28th AIAA Applied Aerodynamics Conference*, (Chicago, Illinois), June 2010.

[116] Qazi, M.-u.-D., He, L., and Permoon, M., "Hammersley sampling and support-vector-regression-driven launch vehicle design," *Journal of Spacecraft and Rockets*, vol. 44, pp. 1094–1106, September-October 2007.

[117] Quinonero, J., Snelson, E., and Wilson, O., "Sensible priors for sparse bayesian learning," Technical Report MSR-TR-2007-121, Microsoft Research, Cambridge, U.K., September 2007.

[118] Rajnarayan, D., Haas, A., and Kroo, I., "A multifidelity gradient-free optimization method and application to aerodynamic design," in *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, no. 2008-6020, (Victoria, British Columbia, Canada), September 2008.

[119] Ran, H. and Mavris, D., "A framework for determination of the weak pareto frontier design solutions under probabilistic constraints," in *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, no. 2006-6960, (Portsmouth, Virginia), September 2006.

[120] Rasmussen, C. E. and Williams, C. K. I., *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[121] Renaud, J. E. and Gabriele, G. A., "Improved coordination in non-hierarchic system optimization," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 31, pp. 2367 – 2373, 1993.

[122] Richard L. Campbell, Melissa B. Carter, O. C. P. D. M. F. L. S., "Design and testing of a blended wing body with boundary layer ingestion nacelles at high reynolds numbers," in *43rd AIAA Aerospace Sciences Meeting and Exhibit*, (Reno, Nevada), January 2005.

[123] Rodriguez, D. L., "Multidisciplinary optimization method for designing boundary-layer-ingesting inlets," *Journal of Aircraft*, vol. 46, pp. 883 – 893, May - June 2009.

[124] Rodriguez, J., Perez, V., Padmanabhan, D., and Renaud, J. E., "Sequential approximate optimization using variable fidelity response surface approximations," *Structural Multidisciplinary Optimization*, vol. 22, pp. 24–34, 2001.

[125] Rodriguez, J., Renaud, J. E., Wujek, B., and Tappeta, R., "Trust region model management in multidisciplinary design optimization," *Journal of Computational and Applied Mathematics*, vol. 124, no. 1-2, pp. 139–154, 2000.

[126] Roman, D., Allen, J., and Liebeck, R., "Aerodynamic design challenges of the blended-wing-body subsonic transport," Tech. Rep. A00-39894, AIAA, 2000.

[127] Roth, B. D. and Kroo, I., "Enhanced collaborative optimization: A decomposition-based method for multidisciplinary design," in *Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, (Brooklyn, New York), ASME, August 2008.

[128] Sack, J., Welch, W., Mitchell, T., and Wynn, H., "Design and analysis of computer experiments," *Statistical Science*, vol. 4, p. 409435, 1989.

[129] Sasena, M. J., Papalambros, P., and Goovaerts, P., "Exploration of metamodeling sampling criteria for constrained global optimization," tech. rep., University of Michigan, 2000.

[130] Sasena, M. J., Papalambros, P., and Goovaerts, P., "Metamodeling sampling criteria in a global optimization," in *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, no. 2000-4921, (Long Beach, California), September 2000.

[131] Schonlau, M., Welch, W. J., and Jones, D. R., "A data-analytic approach to bayesian global optimization," tech. rep., Department of Statistics and Actuarial Science and the Institute for Improvement in Quality and Productivity, University of Waterloo, Ontario, Canada, 1997.

[132] Schlkopf, B. and Smola, A. J., *Learning with Kernels*. MIT Press, 2002.

[133] Sclafani, A. J., DeHaan, M. A., Vassberg, J. C., Rumsey, C. L., and Pulliam, T., "Drag Prediction for the NASA CRM Wing-Body-Tail Using CFL3D and OVERFLOW on an Overset Mesh," in *28th AIAA Applied Aerodynamics Conference*, (Chicago, Illinois), June-July 2010.

[134] Seddon, J. and Goldsmith, E., *Intake Aerodynamics*. AIAA, 2 ed., 1999.

[135] Sellar, R. and Batill, S., "Concurrent subspace optimization using gradient-enhanced neural network approximations," in *6th AIAA/NASA /ISSMO Symposium on Multidisciplinary Analysis and Optimization*, (Reston, VA), September 1996.

[136] Sellar, R., Batill, S. M., and Renaud, J. E., "Response surface based, concurrent subspace optimization for multidisciplinary system design," tech. rep., AIAA, 1996.

[137] Sellar, R., Stelmack, M., Batill, S. M., and Renaud, J. E., "Response surface approximations for discipline coordination in multidisciplinary design optimization," tech. rep., AIAA, 1996.

[138] Serafini, D. B., "A framework for managing models in nonlinear optimization of computationally expensive functions," Tech. Rep. CRPC-TR98781-S, Center for Research on Parallel Computation, Rice University, Houston, Texas, January 1999.

[139] Shan, S. and Wang, G. G., "Survey of modeling and optimization strategies for high-dimensional design problems," in *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Victoria, British Columbia, Canada), September 2008.

[140] Simpson, T. W., Booker, A. J., Ghosh, D., Giunta, A. A., Koch, P. N., , and Yang, R.-J., "Approximation methods in multidisciplinary analysis and optimization: A panel discussion," in *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis & Optimization*, (Atlanta, Georgia), September 2002.

[141] Simpson, T. W., Mauery, T. M., Korte, J. J., and Mistree, F., "Comparison of response surface and kriging models for multidisciplinary design optimization," Tech. Rep. 98-4755, AIAA, 1998.

[142] Simpson, T. W., Peplinski, J. D., Koch, P. N., and Allen, J. K., "Metamodels for computer-based engineering design: Survey and recommendations," *Engineering with Computers*, vol. 17, pp. 129–150, 2001.

[143] Simpson, T. W., Toropov, V., Balabanov, V., and Viana, F. A. C., "Design and analysis of computer experiments in multidisciplinary design optimization: A review of how far we have come or not," in *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Victoria, British Columbia, Canada), September 2008.

[144] Smith, L. H., "Wake ingestion propulsion benefit," *Journal of Propulsion and Power*, vol. 9, pp. 74 – 82, Jan - Feb 1993.

[145] Sobieski, I. P. and Kroo, I., "Collaborative optimization using response surface estimation," Tech. Rep. 98-0915, AIAA, Jan 1998.

[146] Sobieszczanski-Sobieski, J., "Optimization by decomposition: A step from hierarchic to non-hierarchic systems," in *NASA Conference Publication 3031, Part 1, Second NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization*, (Hampton, Virginia), September 1988.

[147] Sobieszczanski-Sobieski, J., Altus, Philips, and Sandusky, "Bilevel integrated system synthesis for concurrent and distributed processing," *AIAA Journal*, vol. 41, no. 10, pp. 1996–2003, 2003.

[148] Sobieszczanski-Sobieski, J. and Haftka, R., "Multidisciplinary aerospace design optimization - survey of recent developments," in *34th Aerospace Sciences Meeting and Exhibit*, (Reno, Nevada), AIAA, January 1996.

[149] Tipping, M., "Bayesian inference: An introduction to principles and practice in machine learning," in *Advanced Lectures on Machine Learning* (Bousquet, O., von Luxburg, U., and Ratsch, G., eds.), pp. 41–62, Springer, 2004.

[150] Tipping, M. E. and Smola, A., "Sparse bayesian learning and the relevance vector machine," 2001.

[151] Toal, D. J., Bressloff, N. W., and Keane, A. J., "Kriging hyperparameter tuning strategies," *AIAA Journal*, vol. 46, pp. 1240–1252, May 2008.

[152] Todoroki, A. and Sekishiro, M., "Modified efficient global optimization for a hat-stiffened composite panel with buckling constraint," *AIAA Journal*, vol. 46, pp. 2257–2264, September 2008.

[153] Torn, A. and Zilinskas, J., eds., *Models and Algorithms for Global Optimization*. Springer, 2007.

[154] Vapnik, V., *Statistical Learning Theory*. Wiley, 1998.

[155] Vavalle, A. and Qin, N., "Iterative response surface based optimization scheme for transonic airfoil design," *Journal of Aircraft*, vol. 44, pp. 365–376, March-April 2007.

[156] Wang, G. G., "Adaptive response surface method using inherited latin hypercube design points," *Journal of Mechanical Design*, vol. 125, pp. 210–220, 2003.

[157] Whitford, R., *Design for Air Combat*. Jane's, 1989.

[158] Winkler, R. L., *An Introduction to Bayesian Inference and Decision*. Holt, Rinehart and Winston, 1972.

VITA

Chung Hyun Lee grew up in Norman, Oklahoma.

- A.B. (bachelor's degree) in Government from Harvard University, 2000
- S.M. (master's degree) in Political Science from Massachusetts Institute of Technology, 2003
- M.S. in Aerospace Engineering from Georgia Institute of Technology, 2007

Between MIT and Georgia Tech, Chung did two years of engineering and mathematics coursework at Boston University to switch careers from political science to engineering.