AIAA SciTech Forum
5-9 January 2015, Kissimmee, Florida
17th AIAA Non-Deterministic Approaches Conference

10.2514/6.2015-0920

Robust Optimization of a Wing Under Structural and Material Uncertainties

Komahan Boopathy [*] Markus P. Rumpfkeil [†]
University of Dayton, Ohio, 45469, USA

Raymond M. Kolonay [‡]
University of Dayton, Ohio, 45469, USA

This paper demonstrates structural sizing optimizations of a fighter wing configuration in the presence of uncertainties in structural parameters and material properties. The design variables and input parameters are considered to have uncertainties and are treated as aleatory and epistemic random variables in the optimization process. The aleatory uncertainties are quantified and propagated via inexpensive sampling of kriging surrogate models, whereas the epistemic uncertainties are propagated using a box-constrained optimization approach. The considered loading condition arises from trimmed flight at a $220^\circ/s$ roll maneuver. The resulting designs are shown to be robust against input anomalies to desired probabilistic levels.

Nomenclature

$\boldsymbol{\alpha}$	aleatory realizations	M	number of variables or dimensions
$\boldsymbol{\beta}$	epistemic realizations	μ	mean
$\boldsymbol{d}$	vector of design variables	N	number of surrogate training points
f	exact function	$\tilde{N}$	number of Monte Carlo samples
f^*	extremum of the function	n	number of exact function evaluations for box-constrained optimization
$\widehat{f}$	surrogate approximated function value	P_k	probability of constraint satisfaction
g	inequality constraint	σ	standard deviation
g^*	extremum of the constraint	σ^2	variance
$\mathcal{J}$	objective function	$\boldsymbol{\xi}$	aleatory random variables
$\frac{d\mathcal{J}}{d\boldsymbol{\xi}}$	aleatory variable gradient	$\boldsymbol{\eta}$	epistemic random variables
$\frac{d\mathcal{J}}{d\boldsymbol{\eta}}$	epistemic variable gradient		

I. Introduction and Motivation

A systematic definition and inclusion of uncertainties in input parameters at the design stage holds the key for producing robust and versatile designs. For example, when designing a system such as an aircraft wing or fuselage, traditionally the structural engineers treat many *characteristic random variables* as constants or fixed, in an attempt to simplify the design process or due to the absence of infrastructure for uncertainty analysis. These random elements influencing the design form a broad spectrum ranging from material properties, operating conditions to manufacturing tolerances. A conventional optimization approach assumes all inputs to be precise and known, which naturally leads to a design that exhibits optimal performance for these exact input settings. Consequently, when such systems encounter off-design conditions, they tend to show degraded performance or can even fail. For example, when the effects of a likely operating environment anomaly such as strong turbulence is not considered upfront, the additional stresses generated

[*] Research Associate, Dept. of Mechanical and Aerospace Eng., KomahanBoopathy@gmail.com, Student Member AIAA
[†] Assistant Professor, Dept. of Mechanical and Aerospace Eng., Markus.Rumpfkeil@udayton.edu, Senior Member AIAA
[‡] Adjunct Professor,Dept. of Mechanical and Aerospace Eng., Raymond.Kolonay@us.af.mil, Associate Fellow AIAA

and acting on the wing can potentially compromise the structural integrity. Similarly, head and tail winds are known to be critical for take-off and landing performance of an airplane. Thus, when a system is designed allowances must be made to accommodate likely variations in the objective function or constraint values that can disrupt the nominal performance. In an optimized design, the optimum solution tends to lie either at the extremum of the objective function or at a constraint boundary.[1] Considering the random elements during the design process, a deterministic optimum can be seen as a vulnerable solution with a large likelihood of violating the design requirements: even small perturbations in the inputs can lead to poor performance or failure of the design. To alleviate some of these problems, a factor of safety is traditionally incorporated into the constraints. The factor of safety serves to move the optimum away from the constraint boundary by a considerable distance, thereby preventing the design from an imminent failure. However, the designer assigning a factor of safety can easily overlook the real effects of uncertainties, and create either over- or under-conservative designs that can lead to weight penalties or vulnerable products, respectively. Moreover, with the continuous evolution of radically new designs, it is increasingly difficult for a designer to assign adequate factors of safety.[1]

The ability to design a system which is less sensitive to randomness in its inputs has, therefore, become the target of recent research efforts. This is accompanied by the emergence and availability of uncertainty analysis procedures and computational resources. Uncertainty quantification (UQ) has grown to be a major field of interest, where the goal is to account for the effect of uncertainties in designs through a modified optimization process known as *optimization under uncertainty* (OUU), where the inputs are treated as random variables. OUU can be subdivided into two fields, namely *robust design optimization* (RDO) and *reliability based design optimization* (RBDO).[2–4] Though these two fields share many common attributes, they differ in their key objectives: RDO techniques minimize the expected mean and variance of the output, whereas the goal of RBDO is to minimize the probability of failure of the system. This work focuses on methods to produce robust designs which involves finding an optimum that is less sensitive to input variability in the objective function or constraint values – as opposed to a deterministic optimum that can exhibit a sharp change in the objective function value for minor perturbations in the inputs. Usually, a robust optimum is obtained at the expense of an increased cost function value compared to a deterministic optimum. Some other goals of OUU are:

- To determine the effects of uncertainties on designs (knowing whether they are robust or vulnerable).
- To identify the limitations of designs and find potential improvements.
- To construct confidence intervals on output quantities that provide valuable information to the designer (e.g., output means and variances).
- To carry out reliability analysis for certification and quality assurance purposes.

"Quantifying uncertainties" is the key to all the potential outcomes including the ones listed above. Uncertainties fall into one of two categories, namely *aleatory* (Type A or reducible) and *epistemic* (Type B or irreducible) uncertainties.[1, 5–9] In recent years, design teams and regulatory agencies are increasingly being asked to specifically characterize and quantify different types of uncertainties and separate their individual effects.[5–9] The most popular and easiest approach for the propagation of uncertainties is the Monte Carlo simulation (MCS) (also known as Monte Carlo sampling), where the simulation output is sampled many times to obtain output statistics or to determine worst case scenarios. However, multiple realizations of the output functions are not always computationally tractable: for example, high-fidelity physics-based simulations such as computational fluid dynamics (CFD) or finite element analyses (FEA) can be very time consuming. To mitigate the problem of exorbitant computational expenses, surrogate models can be constructed to model the uncertainties. Surrogate models are "an approximate but inexpensive to evaluate" representation of the expensive output quantity of interest. They can be sampled exhaustively at a cheaper computational cost to propagate the input uncertainties and determine the output effects. This approach is referred to as the inexpensive Monte Carlo simulation (IMCS).[10] The accuracy of surrogate models is influenced primarily by the choice of training point locations. Training point selection is typically done using design of experiment (DoE) techniques: for example, uniform design,[11] Monte Carlo (MC),[12] latin hypercube (LHS),[13] quadrature nodes,[14, 15] and low-discrepancy sequences.[16] To overcome the difficulties with these methods (e.g., missing important areas, correlated distributions, poorly conditioned linear systems), the authors have recently developed a dynamic training point selection strategy that ably chooses training points in regions that are most viable to improve the accuracy of the surrogate models (see Boopathy and Rumpfkeil[17–19]). This work entails the application of dynamically trained kriging surrogate models for propagating uncertainties in robust structural sizing optimizations.

Outline of the paper

The remainder of this paper is organized as follows. A detailed discussion of the employed robust optimization under uncertainty framework is provided in Section II. The structural optimization of the wing of a fighter airplane is considered as the model application. The current effort focuses on uncertainties of structural and material origin only – other sources of uncertainties are ignored for simplicity. The setup of the optimization problem and the results are discussed in Section III. A summary of important observations are furnished in Section IV.

II. Design in the Presence of Uncertainty

This section provides a detailed discussion of the different stages in optimization under uncertainty,[1,5–9] with a greater emphasis on robust optimization than reliability-based optimization. There are three main stages:

II.A *Identification, modeling and representation of uncertainties*, to translate the available data into mathematical models that are either *probabilistic* or *non-probabilistic* in nature.

II.B *Propagation of uncertainties* using computational models to quantify the impacts on system performance.

II.C *Formulation and solution of the optimization problem* with appropriate objective function and constraints to ensure that the optimum solution is robust against input uncertainties.

II.A. Uncertainty Modeling

The modeling of uncertainties begins with the treatment of appropriate inputs as random variables. Uncertainties can be classified as *aleatory* or *epistemic* uncertainties.[1,5] Aleatory uncertainties are the inherent randomness or variations in physical systems, inputs or operating environments.[5] In other words, the inputs that fluctuate around some mean value with a known probability distribution can be attributed as aleatory random variables. They are inherent because they can not be eliminated (but can be modeled). For example, from a flight operations stand-point, the airline companies possess data for the mean weight and standard deviation of the passengers and their cargo with prescribed probability distributions. Epistemic uncertainties, on the other hand, arise due to the lack of knowledge or information in any phase or activity of the modeling process. It is not an inherent property of the system and thus can be eliminated (or converted to aleatory form) when sufficient data becomes available. Epistemic uncertainties are design inputs which are specified in the form of bounds or intervals (e.g. manufacturing tolerances, range of temperatures during atmospheric re-entries). The underlying probability distributions or other statistical parameters within the interval are unknown.

II.A.1. Probabilistic Modeling

Statistical tools can be used to mathematically model the available data in the form of probability density functions (e.g. Gaussian, log-normal, exponential). The statistical parameters can either be estimated from the available data or in some cases assumed. Figure 1 shows an illustration of fitting available angle of attack information with a normal distribution function. The mean and standard deviation can easily be estimated from the available data and used to define the probability density function (PDF) mathematically as: $p(\xi) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\xi-\mu)^2}{2\sigma^2}}$. When only limited data is available, it is usually prudent to treat the underlying random variable as epistemic rather than aleatory.

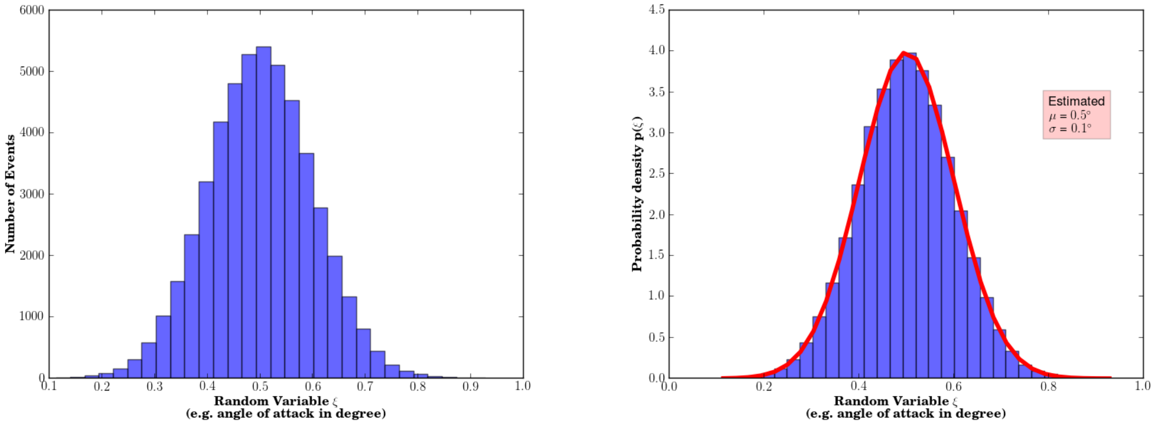


Figure 1: An illustration for the use of probabilistic methods for aleatory uncertainty representation. A histogram of the available angle of attack data (left) is shown with the fitted Gaussian probability density function (right) whose parameters (μ and σ) are estimated from the available data.

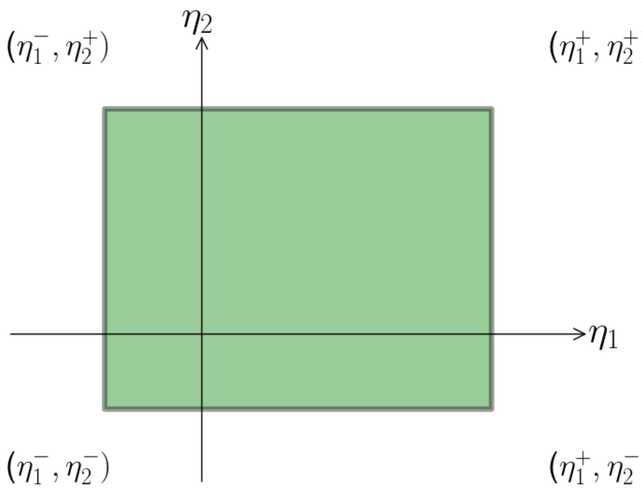


Figure 2: An illustration for interval representation of epistemic uncertainties.

II.A.2. Non-Probabilistic Modeling

The use of probability theory to model the distribution of input uncertainties is not justified, when sufficient information is unavailable. Under these circumstances, non-probabilistic approaches (e.g. possibility theory, interval analysis, convex modeling and evidence theory[1]) can be used for uncertainty modeling. The simplest non-probabilistic approach is the interval representation of input uncertainties. The input random variable is represented by the interval $[\eta^-, \eta^+]$, where η^- and η^+ denotes the lower and upper interval-bounds on the input random variable, respectively. This scenario is illustrated for a two-variable example in Figure 2. The random process can take any value within the specified interval, but the underlying probability distribution is unknown. The goal of the uncertainty propagation via an analysis model (e.g. CFD, FEA) is then to construct intervals for the output quantities of interest based on the input intervals[1] which for non-linear design spaces is non-trivial.

II.B. Uncertainty Propagation

The approaches for the propagation of input uncertainties are discussed next. Having quantified the uncertainties, the next task is to model the input–output relationship through numerical methods. The aleatory variables are denoted as $\boldsymbol{\xi}$ and realizations of aleatory variables from their probability distribution are represented as $\boldsymbol{\alpha}$. The epistemic variables are denoted as $\boldsymbol{\eta}$ and their realizations within the specified interval are denoted as $\boldsymbol{\beta}$.

II.B.1. Propagation of Aleatory Uncertainties

Probabilistic methods which mandate a multitude of realizations are commonly used for computing the output statistics based on the input probability distributions. In other words, the distribution types of the input random variables (e.g., $\boldsymbol{\alpha} \sim \mathcal{N}(\bar{\boldsymbol{\xi}}, \boldsymbol{\sigma}^2_{\boldsymbol{\xi}})$) are known, whereas the functional dependence $f(\boldsymbol{\xi})$ on these random variables is not known, and are modeled using numerical simulations.

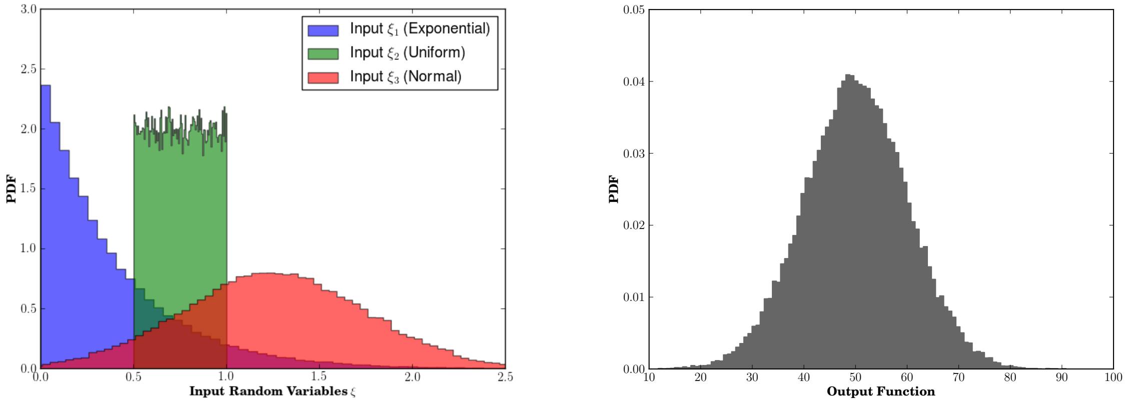
Monte Carlo Simulation: The simplest approach to achieve uncertainty propagation is the Monte Carlo simulation (MCS). When information such as the mean, standard deviation and PDF of the inputs is available, the statistics of the output function can be computed using MCS as demonstrated in Figure 3. In this method, numerous samples (realizations) $\boldsymbol{\alpha}^{(j)}$ are generated from the distribution $p(\boldsymbol{\xi})$ of the input random variables and the response function or simulation code is evaluated for these realizations. This leads to the following estimates for the mean:

$$\bar{f} = \mu_f = \frac{1}{\widetilde{N}} \sum_{j=1}^{\widetilde{N}} f(\boldsymbol{\alpha}^{(j)}), \tag{1}$$

and variance of the output quantity:

$$\sigma_f^2 = \vartheta_f = \frac{1}{\widetilde{N}} \sum_{j=1}^{\widetilde{N}} (f(\boldsymbol{\alpha}^{(j)}) - \bar{f})^2, \tag{2}$$

where $\widetilde{N}$ is the number of Monte Carlo samples. MCS can be used on any output function $f(\boldsymbol{\xi})$ and is hence non-intrusive in nature.



Set of input variables with distributions

Distribution of the output function

Figure 3: An illustration for modeling the input-output relationship during the propagation of aleatory uncertainties.

Inexpensive Monte Carlo Simulation: It is well known that repeated evaluations of the exact function, $f(\boldsymbol{\xi})$, is prohibitively expensive or impractical most of the times. To mitigate this problem, surrogate models can be built and inexpensively probed to yield approximated output function values $\widehat{f}(\boldsymbol{\xi})$ for the calculation of approximate means and variances. The required number of function values (training points) for building a surrogate is usually much smaller than the number required for statistically converging a Monte Carlo simulation. In this work, the domain over which the surrogate model is built is taken to be *three standard* *deviations* in all aleatory input dimensions from the specified mean value *i.e.,* the surrogate domain is: $[\bar{\boldsymbol{\xi}} \pm 3 \cdot \boldsymbol{\sigma}_{\boldsymbol{\xi}}]$. This implies that for normally distributed input variables more than 99 % of all samples (aleatory realizations) fall within the surrogate domain and the less accurate extrapolation capabilities of the surrogate model only need to be employed for a small fraction of the samples. A larger domain can be specified for the surrogate (e.g. 6σ from mean) to account for even remote possibilities which is recommended for reliability calculations.

In this work, kriging surrogate models trained using a dynamic framework developed by Boopathy and Rumpfkeil[17,18] are employed for aleatory uncertainty propagation. The employed kriging surrogate models have the capability to incorporate higher-order derivative information, however, the surrogates here are built using function values only to reduce the complexity in presenting the results.

II.B.2. Propagation of Epistemic Uncertainties

Epistemic uncertainties represent the lack of knowledge about the appropriate value to use:[7] for example, manufacturers typically provide tolerances in terms of intervals (e.g. ± 0.5 *mm* for length of a bolt), – but the exact values are not known and cannot be guaranteed. Here, the goal is to have bounds on the output quantity of interest or to determine worst case scenarios (e.g., maximum possible constraint violation, least possible lift), in order to minimize the sensitivity or variation of the design with respect to these uncertainties. In a situation where only the input intervals $I(\boldsymbol{\eta}) = [\boldsymbol{\eta}^-, \boldsymbol{\eta}^+] = [\bar{\boldsymbol{\eta}} - \boldsymbol{\tau}, \bar{\boldsymbol{\eta}} + \boldsymbol{\tau}]$ are known, the above assessment can be accomplished in a straightforward manner by either one of the two methods discussed below.

Sampling: An extensive sampling of the interval $I(\boldsymbol{\eta})$ can be performed and a simple sorting of the resulting outputs $f(\boldsymbol{\eta})$ can be carried out to determine the extreme values (worst and best cases). However, the computational burden can be prohibitive in the case of high-fidelity physics-based simulations and for higher-dimensional spaces. As a remedy, a surrogate model can be constructed over the domain $I(\boldsymbol{\eta})$ (similar to aleatory uncertainties), which can then be sampled inexpensively. However, with an increasing number of input variables, building an accurate surrogate model requires typically thousands of simulation outputs (referred to as the "curse of dimensionality") and quickly becomes prohibitively expensive as well. In a real life problem there are usually more epistemic than aleatory variables, making surrogate building as well as naive sampling suffer from the "curse of dimensionality".

Box-Constrained Optimization: Alternatively, box-constrained optimizations (BCO)[20,21] can be employed to find the worst and best behavior of the constraint/objective functions within the specified interval $I(\boldsymbol{\eta})$. A gradient-based BCO scales only mildly with the number of input variables, making it computationally more attractive than sampling for quantifying the effect of epistemic uncertainties, particularly for larger problems. In BCO, the problem of finding the extreme value of the function, f^*, (and the constraints, g_i^*) within the interval $I(\boldsymbol{\eta})$ can be cast as follows:

$$\begin{aligned} &\underset{\boldsymbol{\beta}}{\text{minimize/maximize}} && f = f(\boldsymbol{\eta}), \\ &\text{subject to} && \boldsymbol{\beta} \in I(\boldsymbol{\eta}) = [\bar{\boldsymbol{\eta}} - \boldsymbol{\tau}, \bar{\boldsymbol{\eta}} + \boldsymbol{\tau}]. \end{aligned} \tag{3}$$

In most cases the extremum occurs at either the upper or lower bound of the interval due to the quasi-linearity of the usually small space described by $I(\boldsymbol{\eta})$. Thus, BCO typically takes only about five to ten simulation output and gradient evaluations to reach f^* and is used as the method of choice throughout this work to propagate epistemic uncertainties. An L-BFGS[22,23] algorithm which utilizes function and gradient information is used to solve the BCO problem given by Eq. (3).

Figure 4 illustrates the key differences in the treatment of aleatory and epistemic variables in OUU problems.

II.B.3. Propagation of Mixed Uncertainties

Table 1 summarizes four methods that can be employed for the propagation of mixed epistemic and aleatory uncertainties along with their corresponding approximate simulation requirements. The computational requirements can be better interpreted by assuming a typical range of values for: (*i*) the number of Monte Carlo sample points ($\widetilde{N} = 10^5 - 10^8$), (*ii*) the number of surrogate training points ($N = 50 - 5000$), and (*iii*) the number of simulation output evaluations (with gradients) for a BCO ($n = 10 - 100$). The most straightforward way to propagate mixed uncertainties is to carry out a nested-sampling approach (Method

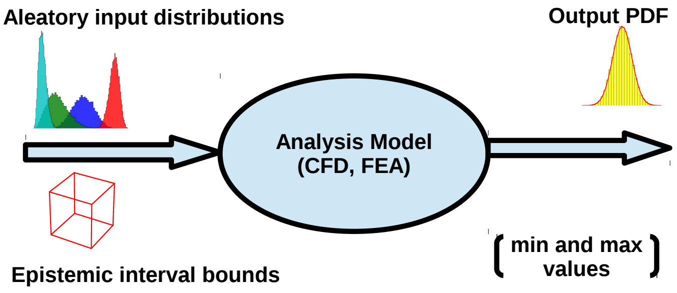


Figure 4: Figure illustrating the propagation of aleatory and epistemic uncertainties.

Table 1: Methods for optimization under mixed uncertainties along with their simulation requirements per iteration.

Method	Propagation Method		No. of Evaluations		Total per iteration
	Aleatory	Epistemic	Aleatory	Epistemic	
1	MCS	MCS	$\widetilde{N}_1$	$\widetilde{N}_2$	$\widetilde{N}_1\widetilde{N}_2$
2	MCS	BCO	$\widetilde{N}$	n	$\widetilde{N} \cdot n$
3	IMCS	IMCS	N_1	N_2	$N_1 \cdot N_2$
4	IMCS	BCO	N	n	$N \cdot n$

1), where for each aleatory random variable realization, $\boldsymbol{\alpha}^{(i)}$, $i = 1, 2, \ldots, \widetilde{N}$, drawn from its probability distribution $p(\boldsymbol{\xi})$, a Monte Carlo sampling (or LHS for a better search performance) has to be performed over the epistemic variable realizations $\boldsymbol{\beta}^{(j)}$, $j = 1, 2, \ldots, \widetilde{N}$, to determine the extreme behavior. Method 2 uses BCO for epistemic uncertainties and is less expensive than Method 1, but can still represent an enormous computational endeavor for the aleatory uncertainties, and is hence impractical for high-fidelity simulations. It can be seen that the last two methods employing surrogate models for uncertainty propagation are several orders of magnitude cheaper. Method 3 turns out to be the cheapest for smaller problems (dimensions less than or equal to 5), but can easily suffer from the curse of dimensionality and thus lacks robustness, whereas it can be inferred that Method 4 is still computationally manageable for bigger problem sizes. Thus, this work employs the IMCS-BCO approach (Method 4) for the propagation of mixed uncertainties in a robust optimization problem. The employed IMCS-BCO framework has been developed by Lockwood *et al.*[20] and Rumpfkeil.[21] A detailed discussion of the steps involved is given in Subsection II.C.3.

The computational requirements in Table 1 are given for just one iteration of the numerical solution of the robust optimization problem. If the optimizer requires $\mathcal{K}$ iterations to converge, the numerical figure in the last column has to be multiplied with $\mathcal{K}$ to obtain an approximation for the number of simulation evaluations needed. As a last remark, a deterministic gradient-based optimization requires on the order of $2\mathcal{K}$ (one function and one gradient evaluation per iteration) simulation output evaluations to reach the optimum.

In the mixed uncertainty problem, the trial design variable vector $\boldsymbol{d}$ used here is comprised of both aleatory and epistemic components *i.e.,* $\boldsymbol{d} = [\bar{\boldsymbol{\xi}}, \bar{\boldsymbol{\eta}}]$, where $\bar{\boldsymbol{\xi}}$ represents the mean of aleatory uncertainties and $\bar{\boldsymbol{\eta}}$ refers to the midpoint of the epistemic uncertainty bounds. The aleatory uncertainties are assumed to be *statistically independent*. Equations for correlated aleatory variables can also be derived; however, the analysis and resulting equations become more complex[24] and are beyond the scope of this work. In addition, the assumed input standard deviation $\boldsymbol{\sigma}$ for aleatory variables as well as the upper and lower bounds of the interval for epistemic variables defined by $\boldsymbol{\tau}$ are treated as fixed throughout the optimization for simplicity.

II.C. Optimization Problem Formulation

II.C.1. Deterministic Optimization

A conventional constrained optimization problem for an objective function, J, that is a function of input variables, $\boldsymbol{d}$, state variables, $\boldsymbol{q}(\boldsymbol{d})$, and simulation outputs, $f(\boldsymbol{d}) = F(\boldsymbol{q}(\boldsymbol{d}), \boldsymbol{d})$, can be written as:

$$\begin{aligned} \underset{\boldsymbol{d}}{\text{minimize}} \quad & J = J(f, \boldsymbol{q}, \boldsymbol{d}), \\ \text{subject to} \quad & R(\boldsymbol{q}, \boldsymbol{d}) = 0, \\ & g(f, \boldsymbol{q}, \boldsymbol{d}) \leq 0. \end{aligned} \tag{4}$$

Here, the state equation residual, R, is expressed as equality constraint, and other system constraints, g, are represented as general inequality constraints. In the case where the input variables are precisely known all functions dependent on $\boldsymbol{d}$ are deterministic. However, in the presence of input uncertainties all functions in Eq. (4) can no longer be treated deterministically.

II.C.2. Robust Optimization

The set up and solution methodologies of the robust OUU problem is discussed next.

Objective Function: A robust objective function, $\mathcal{J}$, can be written in terms of the mean values of the functional outputs μ_{f*} and variance σ^2_{f*}. A robust optimization problem is then given by minimizing the weighted sum of the extremum of the mean and variance of the function. Mathematically, the objective function assumes the form:

$$\mathcal{J} = w_1 \mu_{f*} + w_2\, \sigma^2_{f*}, \tag{5}$$

where w_1 and w_2 are some user specified weights. In this work, the weights w_1 and w_2 are set to one. The asterisk $(*)$ refers to the extremum of the BCO problem.

Constraint Functions: The state equation residual equality constraint, R, is deemed satisfied for all values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The inequality constraints can be cast into a probabilistic statement such that the probability that the constraints are satisfied is greater than or equal to a desired or specified probability, P_k. The constraints are written as a function of mean values and their standard deviations:[25, 26]

$$g^r(\mu_{f*}, \sigma_{f*}, \boldsymbol{q}, \boldsymbol{\xi}, \boldsymbol{\eta}) = g(\mu_{f*}, \boldsymbol{q}, \boldsymbol{\xi}, \boldsymbol{\eta}) + k\sigma_{f*} \leq 0, \tag{6}$$

where k is the number of standard deviations σ_{g*} that the constraint g must be displaced in order to achieve the required P_k.

Problem Formulation: Lastly, the deterministic optimization problem given by Eq. (4) can be recast into a robust design optimization problem[24, 27] as follows:

$$\begin{aligned} \underset{\boldsymbol{\xi}, \boldsymbol{\eta}}{\text{minimize}} \quad & \mathcal{J} = \mathcal{J}(\mu_{f*}, \sigma^2_{f*}, \boldsymbol{q}, \boldsymbol{\xi}, \boldsymbol{\eta}), \\ \text{subject to} \quad & R(\boldsymbol{q}, \boldsymbol{\xi}, \boldsymbol{\eta}) = 0, \\ & g(\mu_{f*}, \boldsymbol{q}, \boldsymbol{\xi}, \boldsymbol{\eta}) + k\sigma_{f*} \leq 0. \end{aligned} \tag{7}$$

Optimization Software: The software package IPOPT (Interior Point OPTimizer)[28] is used for the solution of the robust optimization problem given by Eq. (7). IPOPT also allows users to impose actual bound constraints on the design variables, which can be very helpful in ensuring the robustness of the simulation output analysis by preventing the exploration of too extreme regions of the design space.

II.C.3. Robust Optimization Framework

The steps involved in robust optimization under mixed uncertainties[21] are detailed here (see Figure 5). As described above, surrogate models are built to propagate aleatory uncertainties and box-constrained optimizations are used to propagate epistemic uncertainties.

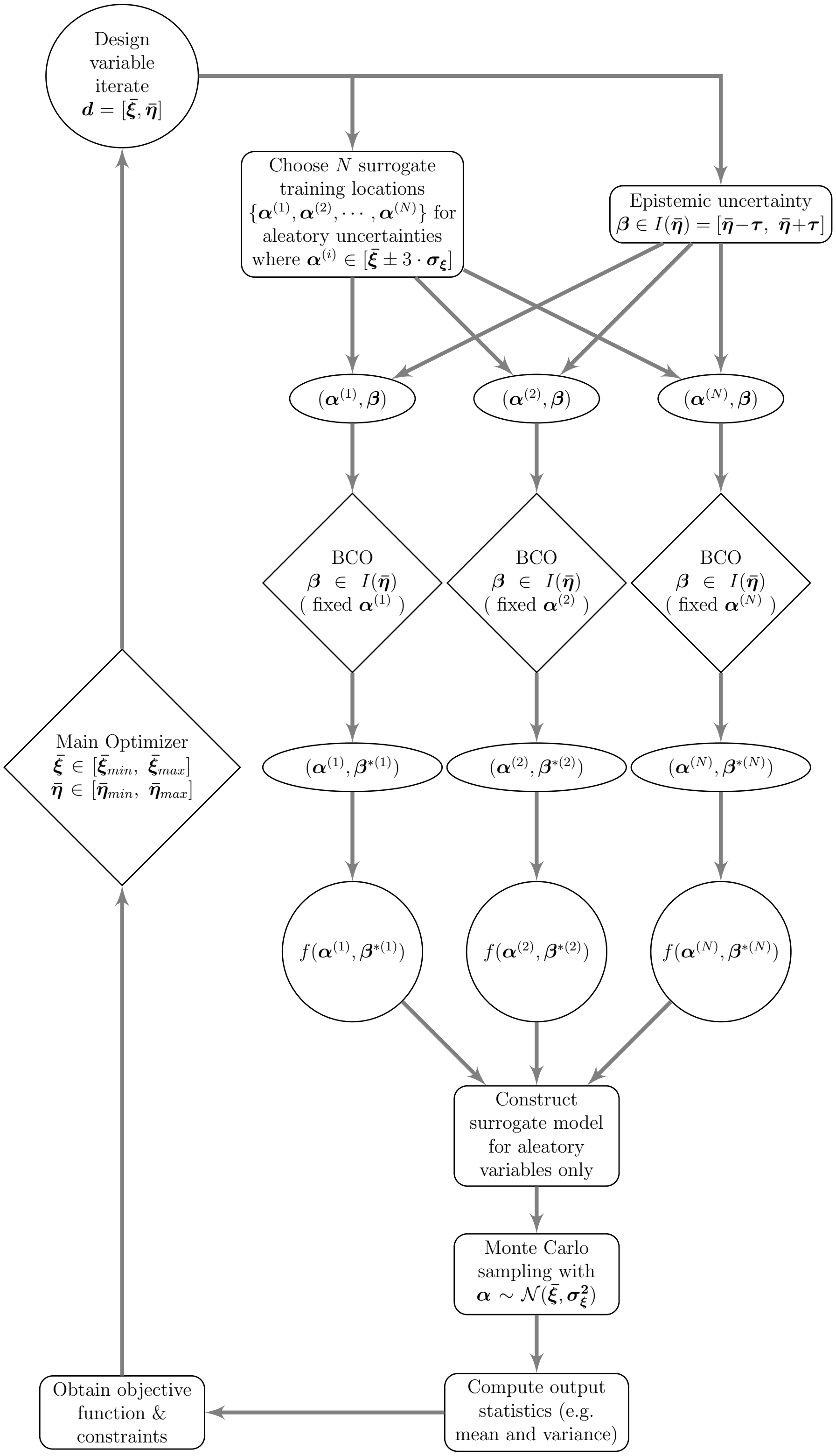

Design variable iterate $\boldsymbol{d} = [\bar{\boldsymbol{\xi}}, \bar{\boldsymbol{\eta}}]$
Choose N surrogate training locations $\{\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}, \cdots, \boldsymbol{\alpha}^{(N)}\}$ for aleatory uncertainties where $\boldsymbol{\alpha}^{(i)} \in [\bar{\boldsymbol{\xi}} \pm 3 \cdot \boldsymbol{\sigma_\xi}]$
Epistemic uncertainty $\boldsymbol{\beta} \in I(\bar{\boldsymbol{\eta}}) = [\bar{\boldsymbol{\eta}} - \boldsymbol{\tau},\ \bar{\boldsymbol{\eta}} + \boldsymbol{\tau}]$
$(\boldsymbol{\alpha}^{(1)}, \boldsymbol{\beta})$
$(\boldsymbol{\alpha}^{(2)}, \boldsymbol{\beta})$
$(\boldsymbol{\alpha}^{(N)}, \boldsymbol{\beta})$
BCO $\boldsymbol{\beta} \in I(\bar{\boldsymbol{\eta}})$ (fixed $\boldsymbol{\alpha}^{(1)}$)
BCO $\boldsymbol{\beta} \in I(\bar{\boldsymbol{\eta}})$ (fixed $\boldsymbol{\alpha}^{(2)}$)
BCO $\boldsymbol{\beta} \in I(\bar{\boldsymbol{\eta}})$ (fixed $\boldsymbol{\alpha}^{(N)}$)
Main Optimizer $\bar{\boldsymbol{\xi}} \in [\bar{\boldsymbol{\xi}}_{min},\ \bar{\boldsymbol{\xi}}_{max}]$ $\bar{\boldsymbol{\eta}} \in [\bar{\boldsymbol{\eta}}_{min},\ \bar{\boldsymbol{\eta}}_{max}]$
$(\boldsymbol{\alpha}^{(1)}, \boldsymbol{\beta}^{*(1)})$
$(\boldsymbol{\alpha}^{(2)}, \boldsymbol{\beta}^{*(2)})$
$(\boldsymbol{\alpha}^{(N)}, \boldsymbol{\beta}^{*(N)})$
$f(\boldsymbol{\alpha}^{(1)}, \boldsymbol{\beta}^{*(1)})$
$f(\boldsymbol{\alpha}^{(2)}, \boldsymbol{\beta}^{*(2)})$
$f(\boldsymbol{\alpha}^{(N)}, \boldsymbol{\beta}^{*(N)})$
Construct surrogate model for aleatory variables only
Monte Carlo sampling with $\boldsymbol{\alpha} \sim \mathcal{N}(\bar{\boldsymbol{\xi}}, \boldsymbol{\sigma_\xi^2})$
Obtain objective function & constraints
Compute output statistics (e.g. mean and variance)

Figure 5: Framework for robust optimization under mixed epistemic and aleatory uncertainties.

1. Initialize: The main optimizer for the OUU problem (IPOPT) takes a trial design vector $\boldsymbol{d} = [\bar{\boldsymbol{\xi}}, \bar{\boldsymbol{\eta}}]$ at each iteration. Based on this design vector, the aleatory surrogate model domain is determined as: $[\bar{\boldsymbol{\xi}} \pm 3 \cdot \boldsymbol{\sigma_\xi}]$.

Within this aleatory domain, surrogate training locations $\boldsymbol{\alpha}^{(i)},\ i = 1, \ldots, N$, are selected using the dynamic training point selection framework.[17,18] For the epistemic counterpart of the design vector, the intervals are determined by $I(\boldsymbol{\eta}) = [\boldsymbol{\eta}^-, \boldsymbol{\eta}^+] = [\bar{\boldsymbol{\eta}} - \boldsymbol{\tau}, \bar{\boldsymbol{\eta}} + \boldsymbol{\tau}]$.

2. Propagate epistemic effects: For each surrogate training point $\boldsymbol{\alpha}^{(i)},\ i = 1, \ldots, N$, a BCO problem is solved for determining the worst or best epistemic realization $\boldsymbol{\beta}^{*(i)}$ within the interval $I(\boldsymbol{\eta})$. Mathematically, this refers to the determination of the extremum f^* of the output function within the interval $I(\boldsymbol{\eta})$:

$$\begin{aligned} &\underset{\boldsymbol{\beta}}{\text{minimize/maximize,}} && f = f(\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}) \\ &\text{subject to} && \boldsymbol{\beta} \in I(\boldsymbol{\eta}). \end{aligned} \tag{8}$$

The aleatory variables remain fixed during the BCO process, while the epistemic variables are allowed to change within the specified interval. This completes the propagation of epistemic uncertainties. Note that the BCO problem only needs a few exact function and gradient evaluations to reach the extremum.

3. Obtain surrogate training data: The exact function $f(\boldsymbol{\xi}, \boldsymbol{\eta})$ is evaluated at $(\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{*(i)}),\ i = 1, \ldots, N$, and the data is used to train the surrogate model. Note that, if the number of aleatory variables is large, the surrogate suffers from the curse of dimensionality, *i.e.,* tens of thousands of BCO results may be required as input training data to the surrogate.

4. Propagate aleatory effects: Once the surrogate model is built using the training data, it can be probed inexpensively to yield the output statistics of the function (e.g. mean μ_{f*} and variance σ^2_{f*}).

5. Update objective/constraints: The aleatory statistics are used to update the objective function and constraints defined in Subsection II.C.2.

Steps (1) to (5) are continued until meeting a user-specified stopping criteria for the robust optimization loop, for example, constraint tolerance, design change tolerance and norm of gradient of the function.

II.C.4. Gradient Evaluation

Aleatory Gradients The gradient of the objective function with respect to design variables associated with aleatory uncertainties (random variables) is given by:[21]

$$\frac{d\mathcal{J}}{d\boldsymbol{\xi}} = \frac{\partial \mathcal{J}}{\partial \mu_{f*}} \frac{d\mu_{f*}}{d\boldsymbol{\xi}} + \frac{\partial \mathcal{J}}{\partial \vartheta_{f*}} \frac{d\vartheta_{f*}}{d\boldsymbol{\xi}} = w_1 \frac{d\mu_{f*}}{d\boldsymbol{\xi}} + w_2 \frac{d\vartheta_{f*}}{d\boldsymbol{\xi}}, \tag{9}$$

where the mean μ_{f*} and variance ϑ_{f*} are computed using the kriging surrogate model. The mean extremum of the simulation output is approximated as:

$$\mu_{f*} \approx \frac{1}{\widetilde{N}} \sum_{k=1}^{\widetilde{N}} \widehat{f^*}(\boldsymbol{\alpha}^k). \tag{10}$$

The derivative of the mean extremum with respect to aleatory variables can be calculated as:

$$\frac{d\mu_{f*}}{d\boldsymbol{\xi}} \approx \frac{1}{\widetilde{N}} \sum_{k=1}^{\widetilde{N}} \frac{d\widehat{f^*}(\boldsymbol{\alpha}^k)}{d\boldsymbol{\alpha}^k} \frac{d\boldsymbol{\alpha}^k}{d\boldsymbol{\xi}} = \frac{1}{\widetilde{N}} \sum_{k=1}^{\widetilde{N}} \frac{d\widehat{f^*}(\boldsymbol{\alpha}^k)}{d\boldsymbol{\alpha}^k}, \tag{11}$$

where $\frac{d\widehat{f^*}(\boldsymbol{\alpha}^k)}{d\boldsymbol{\alpha}^k}$ can be obtained from the kriging surrogate model. Likewise, the variance and its derivative can be approximated as follows:

$$\vartheta_{f*} \quad \approx \quad \left(\frac{1}{\widetilde{N}} \sum_{k=1}^{\widetilde{N}} \widehat{f^*}^2(\boldsymbol{\alpha}^k) \right) - \mu^2_{f*} \tag{12}$$

$$\frac{d\vartheta_{f*}}{d\boldsymbol{\xi}} \quad \approx \quad \left(\frac{2}{\widetilde{N}} \sum_{k=1}^{\widetilde{N}} \widehat{f^*}(\boldsymbol{\alpha}^k) \frac{d\widehat{f^*}(\boldsymbol{\alpha}^k)}{d\boldsymbol{\alpha}^k} \right) - 2\mu_{f*} \frac{d\mu_{f*}}{d\boldsymbol{\xi}}. \tag{13}$$

Epistemic Gradients The gradient of the objective function with respect to design variables associated with epistemic uncertainties (random variables) is given by:

$$\frac{d\mathcal{J}}{d\boldsymbol{\eta}} = \frac{\partial \mathcal{J}}{\partial \mu_{f*}} \frac{d\mu_{f*}}{d\boldsymbol{\eta}} + \frac{\partial \mathcal{J}}{\partial \vartheta_{f*}} \frac{d\vartheta_{f*}}{d\boldsymbol{\eta}} = w_1 \frac{d\mu_{f*}}{d\boldsymbol{\eta}} + w_2 \frac{d\vartheta_{f*}}{d\boldsymbol{\eta}}. \tag{14}$$

In this case, the calculation of $\frac{d\mu_{f*}}{d\boldsymbol{\eta}}$ and $\frac{d\vartheta_{f*}}{d\boldsymbol{\eta}}$ is not trivial: moving the midpoint of the epistemic intervals will lead in general to different extrema for the training points and thus to a different surrogate model, which when sampled provides different values for μ_{f*} and ϑ_{f*}. In comparison, the aleatory gradient was easy to obtain because the same surrogate model is used and only the change in sample points (random realizations) has to be accounted for. In this work the following approximations are used:[21]

$$\frac{d\mu_{f*}}{d\boldsymbol{\eta}} \approx \left.\frac{df}{d\boldsymbol{\eta}}\right|_{(\boldsymbol{\xi}=\bar{\boldsymbol{\xi}},\boldsymbol{\eta}=\bar{\boldsymbol{\eta}})} \quad \text{and} \quad \frac{d\vartheta_{f*}}{d\boldsymbol{\eta}} \approx 0, \tag{15}$$

i.e., the derivative of the mean extremum μ_{f*} with respect to the epistemic variables $\boldsymbol{\eta}$, is approximated by the derivative of the function f with respect to $\boldsymbol{\eta}$, evaluated at the mean values of the aleatory variables and midpoints of the interval for the epistemic variables. Generally, this derivative is non-zero: since for the epistemic optimizations via BCO, the extreme value is typically encountered at the interval bound. Since the variances are small in comparison with the mean values, their sensitivities are neglected: $\frac{d\vartheta_{f*}}{d\boldsymbol{\eta}} \approx 0$.

III. Wing Optimization

III.A. Analysis Software and Model

ASTROS (Automated Structural Optimization System) is a finite element analysis and optimization software developed at the Air Force Research Laboratory for the preliminary design of aircraft structures.[29–32] ASTROS has built-in analysis and optimization capabilities, however in this work ASTROS is used only for structural analysis *i.e.,* it supplies the objective function, constraint values and their corresponding sensitivities to the robust optimization framework described in the previous section. An in-house Fortran-Python interface is used to obtain these values for the given set of input parameters from the produced CADDB database.[29]

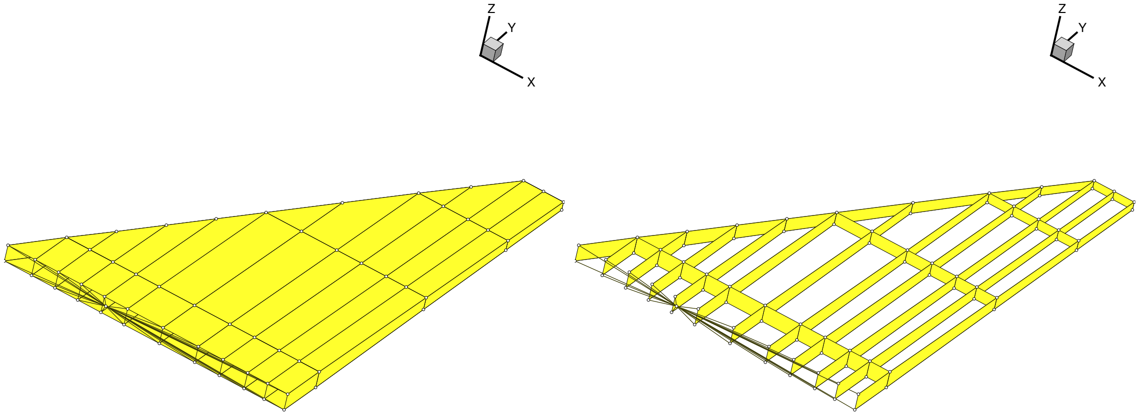


Figure 6: Structural analysis model of a fighter wing used in optimization.

The structural analysis model of a fighter wing is shown in Figure 6. In this work, optimization is demonstrated for the structural components only; robust aero-structural optimization under uncertainty falls beyond the scope of this work. The structural components of the wing and their associated finite element types are listed in Table 2.

For the analyses, trimmed flight at 220°/s roll maneuver at a Mach number of 0.7 and a dynamic pressure of 5.86 *psi* is considered as the loading condition.

III.B. Optimization Problem

The setup of the deterministic optimization problem is described next, followed by the robust OUU problem. The deterministic objective is to design a minimum weight wing structure by optimizing the thickness/cross-sectional area of the structural members (listed in Table 2), while meeting user-defined constraints (listed in

Table 2: Components of the wing analysis model with corresponding element types.

Wing component	Element Type	Design Variable ID	Lower Limit	Upper Limit
Connection Rods for Shear Elements	PROD	1	0.10 in^2	10.0 in^2
Spars	PSHEAR	2, 3, 4, 5, 19, 20, 21, 22	0.25 in	1.50 in
Spar Caps	PROD	6, 7, 8, 9, 23, 24, 25, 26	0.10 in^2	1.25 in^2
Ribs	PSHEAR	10	0.25 in	1.50 in
Skins	PQDMEM1/ PTRMEM1	11, 12, 13, 14, 15, 16, 17, 18	0.10 in	1.50 in

Table 3). Mathematically, the optimization problem can be written as:

$$\begin{aligned} \underset{\boldsymbol{d}}{\text{minimize}} \quad & W = W(\boldsymbol{d}), \\ \text{subject to} \quad & g_{disp} = \frac{Z}{Z_{max}} - 1 \leq 0, \\ & g_{stress} = \frac{\Sigma}{\Sigma_{max}} - 1 \leq 0, \\ & \boldsymbol{d}_{lb} \leq \ \boldsymbol{d}_{1-26} \ \leq \boldsymbol{d}_{ub}, \end{aligned} \tag{16}$$

where Σ refers to the von Mises stresses and Z refers to the vertical nodal displacements at the aft end of the wing.

Table 3: List of constraints in the optimization problem.

Constraint Type	Description	Symbol	Quantity	Value
Displacement	Wing tip (6 nodes)	g_{1-6} g_{7-12}	Upper limit Lower limit	$+3.0$ in -3.0 in
von Mises Stress	Top skins (28)	g_{13-40}	Tensile limit(13-21) Compression limit(22-30) Shear limit(32-40)	$+1.0 \cdot 10^4$ psi $-1.0 \cdot 10^4$ psi $+5.0 \cdot 10^3$ psi
von Mises Stress	Bottom skins (28)	g_{41-68}	Tensile limit(41-49) Compression limit(50-59) Shear limit(60-68)	$+1.0 \cdot 10^4$ psi $-1.0 \cdot 10^4$ psi $+5.0 \cdot 10^3$ psi

In the robust optimization problem, the inputs are considered to have uncertainties and are treated as random variables. The type of uncertainties attributed with these inputs are detailed in Table 4: the structural entities are assumed to have epistemic uncertainties, whereas the material properties of Aluminum are assumed to have aleatory uncertainties with Gaussian distributions. Many other input parameters such as Mach number and roll rate could also be treated as random variables, but they are not considered here for simplicity. The deterministic problem is transformed into a robust optimization problem by minimizing the

Table 4: Assumed input uncertainties for the wing optimization under uncertainty problem.

Random	Symbol	Uncertainty	Distribution	Lower	Upper	Mean	Std.	Unit
Variable		Type	Type	Bound	Bound		Dev.	
Skins, spars, spar caps ribs, posts	$\boldsymbol{d}_{1-26}$	Epistemic	–	-0.025	0.025	–	–	in
Young's modulus	E	Aleatory	Normal	–	–	10^7	$2.5 \cdot 10^4$	psi
Poisson ratio	ν	Aleatory	Normal	–	–	0.33	0.033	–
Weight density	ρ	Aleatory	Normal	–	–	0.10	0.003	lb/in^3

equally weighted sum of mean and variance of the weight, while the constraints are moved user-specified k-standard deviations away from their mean values (in the direction of decreasing constraint value). Mathematically,

$$\begin{aligned} &\text{minimize} \quad && \mathcal{J} = \mu_W + \sigma_W^2, \\ &\text{subject to} \quad && g_i^r = \mu_{g_i} + k\sigma_{g_i} \leq 0, \quad \text{for } i = 1, \ldots, 68, \end{aligned} \tag{17}$$

where i refers to the stress or displacement constraints that are listed in Table 3. The total number of constraints in the optimization problem is 68.

From a robust OUU stand-point, this implies that for each of these constraints and the objective function, a kriging surrogate model needs to be built (as described in Section II) and sampled to yield the necessary statistics (output mean and standard deviation/variance) to update Eq. (17).

III.C. Optimization Results

The results of the deterministic and robust wing optimizations are discussed next. A total of 6 cases were considered: one deterministic and five robust optimizations corresponding to $k = 0\ ,1,\ 2,\ 3,\ 4$, for all of which IPOPT has been used as the optimization software.

III.C.1. Objective Function

The design variable values at the optimum solution for different optimization cases are listed in Table 5. One can see a clear difference in the optimized values for the deterministic vs non-deterministic approach especially for design variables one, ten, thirteen to nineteen, and twenty-three.

Table 5: The design variable values at the initial and optimum designs.

DV	Initial	Deterministic	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$
1	5.050	0.617	0.699	0.701	0.704	0.710	0.711
2	0.875	0.252	0.254	0.254	0.254	0.255	0.255
3	0.875	0.250	0.251	0.251	0.251	0.251	0.251
4	0.875	0.252	0.251	0.251	0.251	0.251	0.251
5	0.875	0.260	0.260	0.259	0.258	0.258	0.257
6	0.675	0.104	0.108	0.107	0.109	0.108	0.108
7	0.675	0.100	0.100	0.100	0.100	0.100	0.100
8	0.675	0.102	0.102	0.102	0.102	0.102	0.102
9	0.675	0.104	0.104	0.104	0.104	0.104	0.104
10	0.875	0.354	0.409	0.412	0.413	0.416	0.422
11	0.800	0.111	0.111	0.111	0.111	0.111	0.111
12	0.800	0.128	0.133	0.134	0.134	0.132	0.133
13	0.800	0.342	0.395	0.397	0.397	0.405	0.406
14	0.800	0.386	0.428	0.421	0.413	0.435	0.436
15	0.800	0.166	0.210	0.212	0.213	0.217	0.219
16	0.800	0.265	0.324	0.324	0.325	0.325	0.325
17	0.800	0.519	0.581	0.588	0.594	0.600	0.605
18	0.800	0.405	0.443	0.449	0.458	0.456	0.463
19	0.875	0.276	0.297	0.300	0.304	0.311	0.319
20	0.875	0.257	0.256	0.255	0.254	0.254	0.253
21	0.875	0.324	0.360	0.365	0.366	0.372	0.376
22	0.875	0.316	0.347	0.347	0.349	0.354	0.361
23	0.675	0.332	0.434	0.440	0.451	0.455	0.458
24	0.675	0.101	0.101	0.101	0.101	0.101	0.101
25	0.675	0.104	0.107	0.106	0.106	0.106	0.106
26	0.675	0.113	0.116	0.117	0.118	0.119	0.121

Table 6 compares the objective function values of the deterministic design with those of the robust designs. By observing the optimum weights, it can be inferred that the robust design is heavier than the deterministic design, whose percentage increases are given in the last two column. The deterministic design is feasible and has the lowest cost function value $\mathcal{J}$, but is highly likely to violate one or many of the stress and displacement requirements, when the parameters are deviant from the expected values because of uncertainties. This likelihood is reduced for the robust solutions, to probability levels specified in column P_k, because the thicknesses (area for the first design variable) of the structural design components and their material properties are treated as random variables upfront and their effects are analyzed within the IMCS-BCO approach. The penalty for the more robust designs is that the optimizer has to add more weight to the structure. Note that the fixed weight of the structure which is the part that is not designed, for example, fuel, payload, tip missile, etc., is not included in the objective function directly and is 24360 *lb*. Thus, the total weight is the sum of the objective function and this fixed weight.

Table 6: Summary of results pertaining to the objective function values for deterministic and robust optima.

Type	k	P_k	μ_W	σ_W^2	$\mathcal{J}$	Total Structural	% Increase in	% Increase
			lb	*lb*	*lb*	Weight *lb*	Cost Function	Total Weight
Deterministic	-	-	103.7	-	103.7	24463.7	-	-
Robust	0	0.5000	144.3	18.7	163.0	24504.3	39.2	0.166
Robust	1	0.8413	144.8	18.9	163.7	24504.8	39.6	0.168
Robust	2	0.9772	145.4	19.0	164.4	24505.4	40.2	0.170
Robust	3	0.9986	146.0	19.2	165.2	24506.0	40.8	0.173
Robust	4	0.9999	146.5	19.3	165.8	24506.5	41.3	0.175

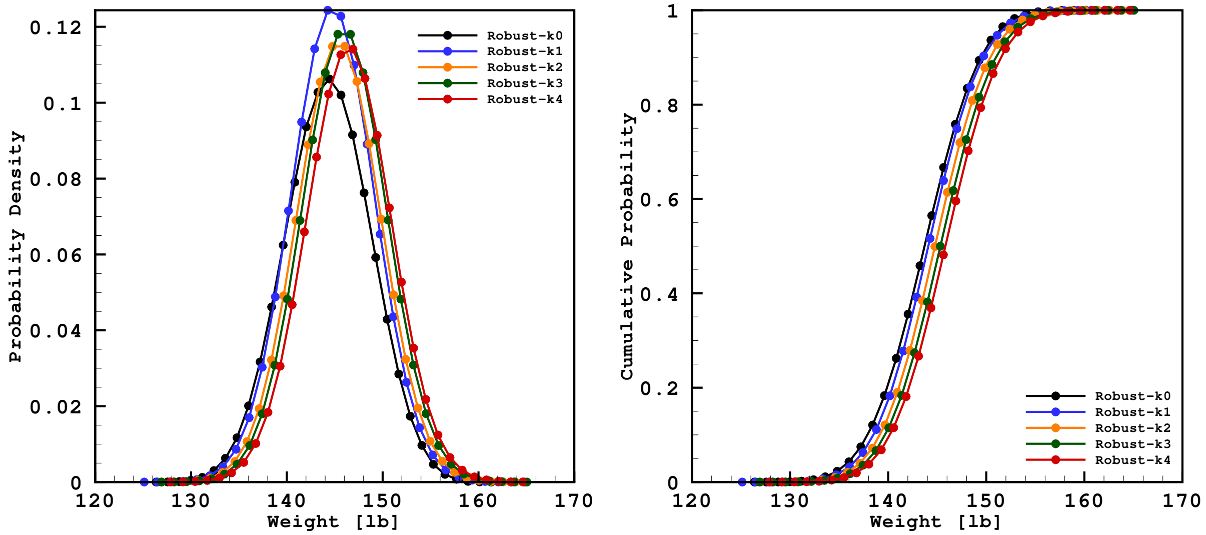


Figure 7: PDFs (left) and CDFs (right) of the objective function for different robust cases.

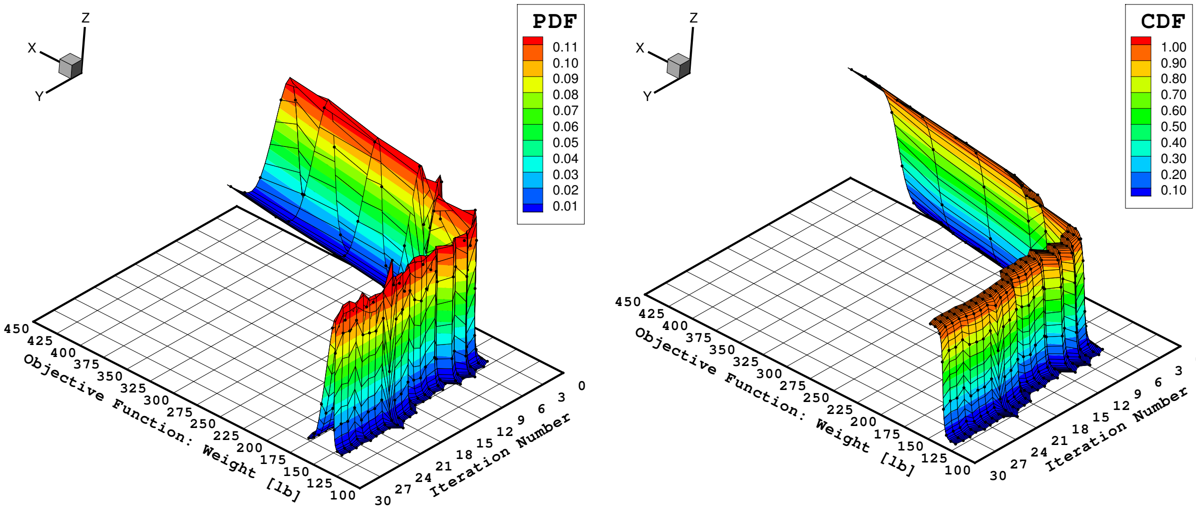


Figure 8: Plot of change in PDF (left) and CDF (right) of the objective function with the number of optimizer iterations ($k = 4$).

Figure 7 depicts the PDF and CDF of the objective function at the optimum design for the five different robust cases. Figure 8 provides a visualization of the same at different stages (iterations) in the optimization for $k = 4$. These PDFs and CDFs can give an idea of the possible spread of values as well as probabilities to the designer. Figure 9 shows the optimization convergence history of the objective function. One can notice the inability of the optimizer to make further progress in the last 10 or so iterations.

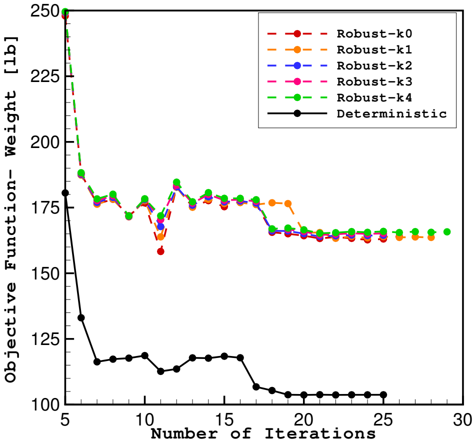


Figure 9: Plot of change in objective function with the number of optimizer iterations.

III.C.2. Constraints

Table 7 provides the list of constraints that are active at the optimum solution out of the total 68 constraints. These constraints have a greater impact on the design. It can be noticed that the deterministic design has more active constraints than the robust ones.

Table 7: List of constraints that are active at the optimum solution: $|g_i| < 10^{-2}$.

Opt. Case	Deterministic	Robust-k0	Robust-k1	Robust-k2	Robust-k3	Robust-k4
# of active constraints	10	1	4	2	1	1
List	31,32 47,48,49,50 55,56,57,58	57	47,48 57, 58	32 57	57	57

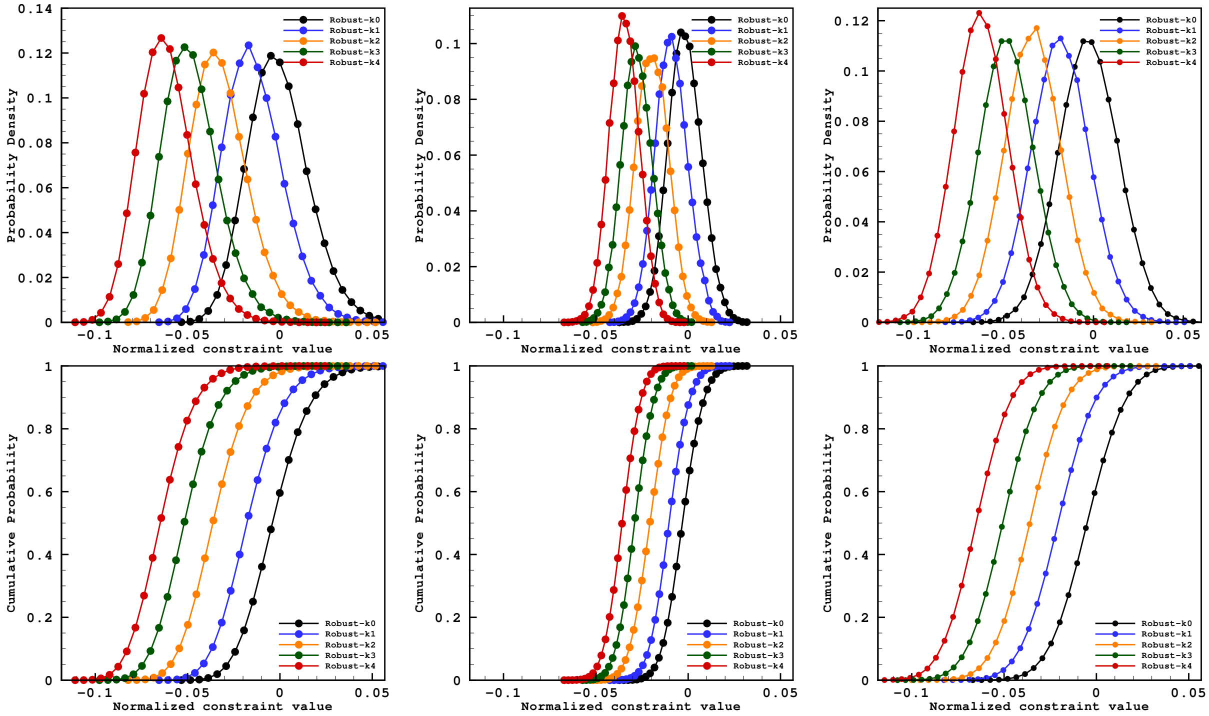
Figure 10 shows the PDF and CDF of selected constraints. The designer can assess the probability that the given constraint is satisfied from the CDFs as well as the spread of potential outcomes from the PDFs. The robust optimization tends to move the constraints in the direction of greater constraint compliance, which can be observed from the plots corresponding to different k values. This helps produce a design that is less prone to failure due to the effect of uncertainties. The probability that a constraint will be violated for a design corresponding to $k = 0$ is 50%, whereas this probability reduces and becomes negligible as k value increases. The requirement for a greater constraint compliance adds more penalty to the objective function though, as shown in Table 6. Figure 11 plots the distributions of selected constraints for $k = 4$ with the number of optimizer iterations.

Figure 12 compares the nodal displacements for the deterministic and the robust design corresponding to $k = 4$. Though displacement constraints did not actively govern the optimization process, it can be inferred from the figure that the robust design leads to less nodal displacement.

Figure 13 shows the spanwise vertical displacement plotted along the leading and trailing edges of the wing box. The deterministic design has higher displacement values than the robust one.

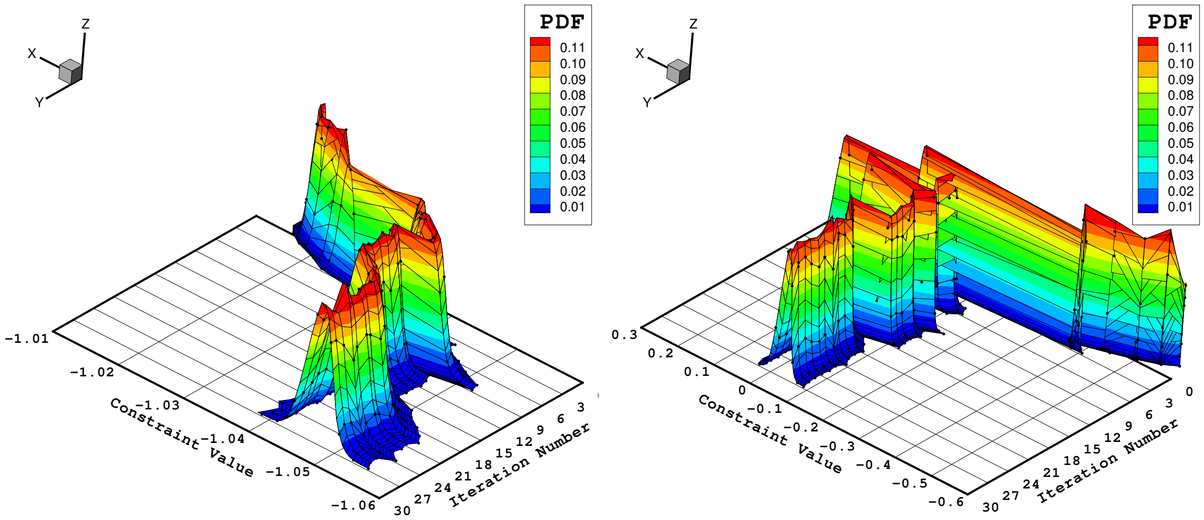
III.C.3. Verifications and Computational Cost

Here, verifications for the IMCS-BCO approach are provided by a selective comparison of the $k = 4$ case with exact Monte Carlo simulation (MCS) and BCO, *i.e.,* the surrogate models are replaced with exact FEA



g_{32} g_{47} g_{58}

Figure 10: Comparison of PDFs (top) and CDFs (bottom) different robust design for selected constraints.



Displacement - g_1 Compressive stress - g_{57}

Figure 11: Change in PDFs with the number of iterations for selected constraints ($k = 4$).

solves. Only 10000 Monte Carlo samples are used for this test. For each Monte Carlo sample drawn from the input aleatory distribution, a BCO problem is solved and statistics obtained are presented in Table 8.

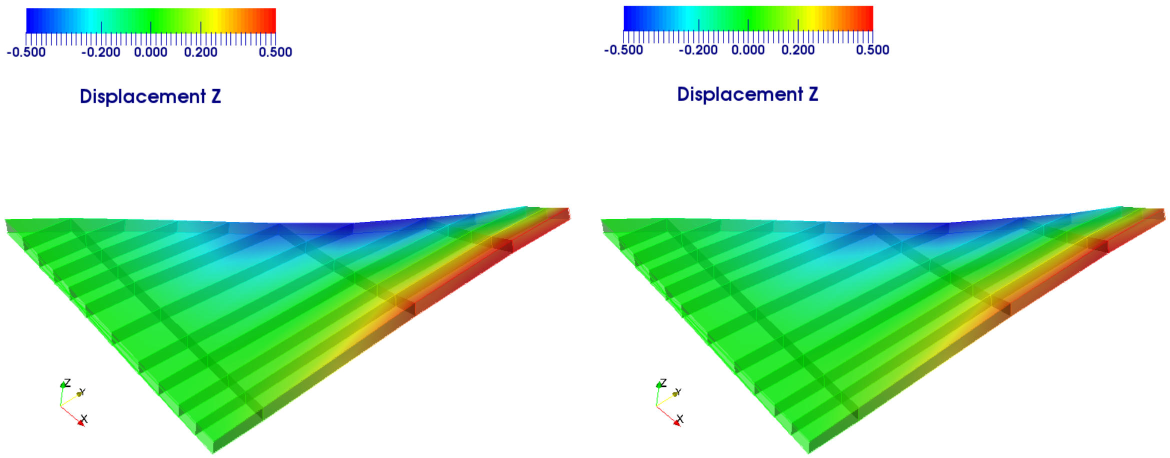


Figure 12: Nodal displacements in vertical direction for deterministic (left) and $k = 4$ robust designs (right). Note that warping has been done for better visualizations with a scale factor of 15.

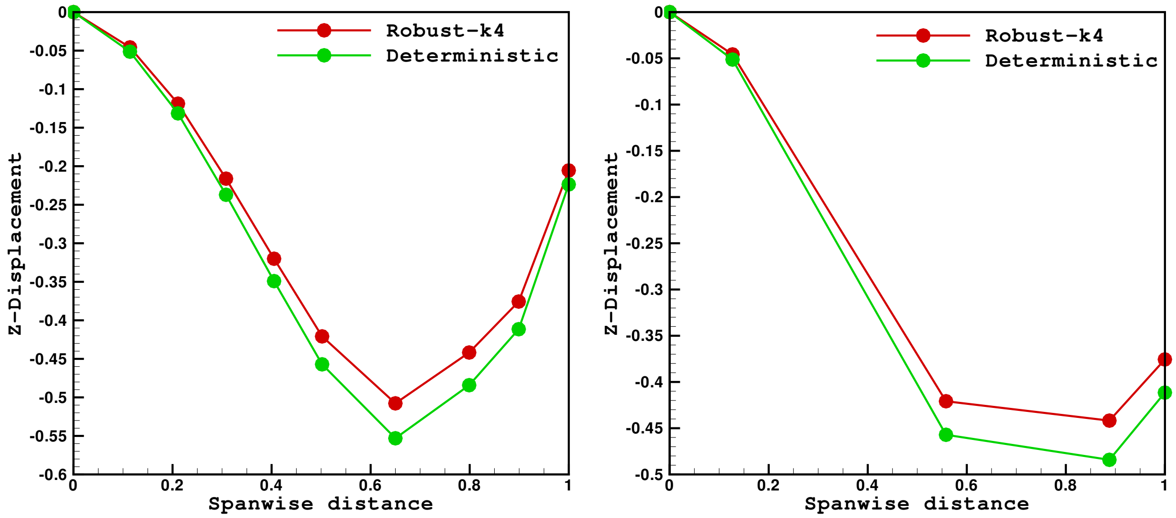


Figure 13: Plot of spanwise nodal displacements along the leading (left) and trailing edges (right).

Table 8: Comparison of IMCS-BCO with MCS-BCO for mixed OUU propagation.

Function	Simulation Type	μ_W	σ_W^2	No. of ASTROS calls
Weight	IMCS-BCO	405.2625	147.8309	61
	MCS-BCO	405.4212	148.1853	31322

Table 9 summarizes the number of FEA solutions (*i.e.*, ASTROS calls) to yield function and gradient values as needed, as well as the CPU time taken to solve each one of these OUU problems. A total of 69 surrogate models were built in every robust optimization iteration with 20 training points each. A BCO problem needs to be solved for each training point for the surrogates. Though the number of FEA solutions are quite high, it is noteworthy that the surrogate models produce sufficiently accurate representation of the objective function/constraint values, for a fraction of computational cost as shown in Table 8. Without the use of surrogate models for aleatory uncertainties, a naive MCS would have increased the number of FEA solves by a factor of $\widetilde{N}/20$.

Table 9: A comparison of computational cost for robust and deterministic optimizations.

Opt. Case	Deterministic	Robust-k0	Robust-k1	Robust-k2	Robust-k3	Robust-k4
CPU Hours	0.15	353.3	394.1	343.9	343.5	402.0
Avg. # F/FG per surrogate (including BCOs)	-	189	190	189	189	189
Avg. # F/FG per OUU iteration	69	13010	13073	13020	12998	13011
No. of optimizer iterations	26	27	29	26	26	30
Total # of F/FG Evals.	1794	351270	379110	338504	337941	390327

IV. Conclusion

This paper demonstrated a robust optimization framework under mixed epistemic and aleatory uncertainties using surrogate models for an application of interest to aircraft structural engineers. Robust optimizations were carried-out using the IMCS-BCO framework for different user specified k values (0 to 4) and the designs were compared with the deterministic solution. The number of function and gradient evaluations needed for deterministic and robust optimizations were included, with an emphasis on computational savings by the use of surrogate models. Robustness studies in terms of PDFs and CDFs of the objective function and constraints were presented and discussed. Finally, verifications for the propagation of aleatory uncertainties were shown by selective comparisons with full Monte Carlo simulations.

Acknowledgments

The authors thank Wataru Yamazaki for his kriging surrogate model. This work was supported in part by an allocation of computing time from the Ohio Supercomputer Center.

References

[1] Keane, A. and Nair, P., *Computational Approaches for Aerospace Design*, John Wiley & Sons, 2005.

[2] Arora, J. S., *Optimization of Structural and Mechanical Systems*, World Scientific Publishing Co. Pte. Ltd., 2007.

[3] Agarwal, H., *Reliability Based Design Optimization: Formulations and Methodologies*, Ph.D. thesis, University of Notre Dame, 2004.

[4] Padmanaban, D., *Reliability-Based Optimization for Multidisciplinary System Design*, Ph.D. thesis, University of Notre Dame, 2003.

[5] Alvin, K. F., Oberkampf, W. L., Rutherford, B. M., and Diegert, K. V., "Methodology for Characterizing Modeling and Discretization Uncertainties in Computational Simulations," Tech. Rep. SAND2000-5015, Sandia National Laboratories, 2000.

[6] Pilch, M., Trucano, T. G., and Helton, J. C., "Ideas Underlying Quantification of Margins and Uncertainties (QMU): A white paper," Tech. Rep. SAND2006-5001, Sandia National Laboratories, 2006.

[7] Helton, J. C., Johnson, J. D., Oberkampf, W. L., and Storlie, C. B., "A sampling-based computational strategy for the representation of epistemic uncertainty in model predictions with evidence theory," Tech. Rep. SAND2006-5557, Sandia National Laboratories, 2006.

[8] Diegert, K., Klenke, S., Novotny, G., Paulsen, R., Pilch, M., and Trucano, T., "Toward a More Rigorous Application of Margins and Uncertainties within the Nuclear Weapons Life Cycle - A Sandia Perspective," Tech. Rep. SAND2007-6219, Sandia National Laboratories, 2007.

[9] Helton, J. C., Oberkampf, J. D. J. W. L., and Sallaberry, C. J., "Representation of Analysis Results Involving Aleatory and Epistemic Uncertainty," Tech. Rep. SAND2008-4379, Sandia National Laboratories, 2008.

[10] Ghate, D. and Giles, M. B., "Inexpensive Monte Carlo uncertainty analysis," *Recent Trends in Aerospace Design and Optimization, Tata McGraw-Hill, New Delhi*, 2006, pp. 203–210.

[11] Fang, K. T., Lin, D., and Winker, P., "Uniform design: Theory and application," *Technometrics*, Vol. 42, No. 3, 2000, pp. 237–248.

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

[13] McKay, M. D., Conover, W. J., and Beckman, R. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–245.

[14] Xiu, D., *Numerical Methods for Stochastic Computations: A Spectral Method Approach*, Princeton University Press.

[15] Maître, O. P. L. and Knio, O. M., *Spectral Methods for Uncertainty Quantification, Scientific Computation*, Springer.

[16] Wong, T. T., Luk, W. S., and Heng, P. A., "Sampling with Hammersley and Halton points," *J. Graph. Tools*, Vol. 2, No. 2, Nov. 1997, pp. 9–24.

[17] Boopathy, K. and Rumpfkeil, M. P., "Unified Framework for Training Point Selection and Error Estimation for Surrogate Models," *AIAA Journal*, 2014. http://dx.doi.org/10.2514/1.J053064.

[18] Boopathy, K. and Rumpfkeil, M. P., "A Multivariate Interpolation and Regression Enhanced Kriging Surrogate Model," AIAA Paper, 2013-2964.

[19] Boopathy, K. and Rumpfkeil, M. P., "Building Aerodynamic Databases Using Enhanced Kriging Surrogate Models," AIAA Region III Student Conference, Illinois Institute of Technology, Chicago, 2013.

[20] Lockwood, B., Anitescu, M., and Mavriplis, D. J., " Mixed Aleatory/Epistemic Uncertainty Quantification for Hypersonic Flows via Gradient-Based Optimization and Surrogate Models," AIAA Paper, 2012-1254, 2012.

[21] Rumpfkeil, M. P., "Optimizations Under Uncertainty Using Gradients, Hessians, and Surrogate Models," *AIAA Journal*, Vol. 51, No. 2, 2013, pp. 444–451.

[22] Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C., "A Limited Memory Algorithm for Bound Constrained Optimization," *SIAM Journal on Scientific Computing*, Vol. 16(5), 1995, pp. 1190–1208.

[23] Zhu, C., Byrd, R. H., Lu, P., and Nocedal, J., "L-BFGS-B: A Limited Memory FORTRAN Code for Solving Bound Constrained Optimization Problems," Tech. Rep. NAM-11, Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, Illinois, USA, 1994.

[24] Putko, M. M., Newmann, P. A., Taylor III, A. C., and Green, L. L., "Approach for uncertainty propagation and robust design in CFD using sensitivity derivatives," AIAA Paper, 2001-2528, June, 2001.

[25] Du, X. and Chen, W., "Methodology for Managing the Effect of Uncertainty in Simulation-Based Design," *AIAA Journal*, Vol. 38(8), 2000, pp. 1471–1478.

[26] Parkinson, A., Sorensen, C., and Pourhassan, N., "A general approach for robust optimal design," *Trans. ASME*, Vol. 115, 1993, pp. 74–80.

[27] Putko, M. M., Taylor III, A. C., Newmann, P. A., and Green, L. L., "Approach for Input Uncertainty Propagation and Robust Design in CFD Using Sensitivity Derivatives," *Journal of Fluids Engineering*, Vol. 124(1), 2002, pp. 60–69.

[28] Waechter, A. and Biegler, L. T., "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Mathematical Programming*, Vol. 106(1), 2006, pp. 25–57.

[29] Johnson, E. H. and Venkayya, V. B., *ASTROS Theoretical Manual, AFWAL - TR - 88 - 3028*, Wright Patterson Air Force Base, Dayton, OH, 1988.

[30] Neill, D. J. and Herendeen, D. L., *ASTROS User Manual, WL - TR - 93 - 3025*, Wright Patterson Air Force Base, Dayton, OH, 1993.

[31] Neill, D. J., Herendeen, D. L., and Hoesly, R. L., *ASTROS Programmer's Manual, WL - TR - 93 - 3038*, Wright Patterson Air Force Base, Dayton, OH, 1988.

[32] Johnson, E. H. and Neill, D. J., *ASTROS Applications Manual, AFWAL - TR - 88 - 3028*, Wright Patterson Air Force Base, Dayton, OH, 1988.